



Provided by the author(s) and University of Galway in accordance with publisher policies. Please cite the published version when available.

Title	Contributions towards 3D synthetic facial data generation and 3D face analysis with weak supervision
Author(s)	Basak, Shubhajit
Publication Date	2024-03-11
Publisher	NUI Galway
Item record	http://hdl.handle.net/10379/18085

Downloaded 2024-04-27T15:24:22Z

Some rights reserved. For more information, please see the item record link above.



Contributions Towards 3D Synthetic Facial Data Generation And 3D Face Analysis With Weak Supervision



OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Shubhajit Basak

Supervisors: Dr. Michael Schukat (Primary)

Dr. Rachel McDonnell (Secondary)

Advisor: Prof Peter Corcoran

School Of Computer Science
University of Galway

This dissertation is submitted for the degree of
Doctor of Philosophy

March 2024

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others except as specified in the text and Acknowledgements.

Shubhajit Basak
March 2024

Acknowledgements

I want to express my sincere gratitude to my supervisor Dr. Michael Schukat for his immense support, guidance, and encouragement. He always supported me and made me believe I could accomplish my Ph.D. journey. In the initial days of my Ph.D. journey, his positivity and continuous encouragement helped me to come out of my imposter syndrome. He has been there whenever I reached out for help during my research and writing of this thesis. He was always patient with me even in my most nonproductive days.

A very special thanks to my mentor Prof Peter Corcoran for his valuable feedback, encouragement, and believing in me. Though he was not my official supervisor, still he always had time for me to discuss my ideas. He was always there to arrange for any infrastructure and other requirements to accomplish my research experiments. With his sound guidance, I felt motivated and encouraged at every step of this journey, and feel blessed to have a mentor like him.

I am indebted to Prof Christopher Dainty for his help in my early days. I still remember the day he called me into his office and showed me how to start looking for research papers and identifying research groups. As I do not have any previous research experience that particular discussion helped me a lot to overcome those initial challenging days. My sincere thanks to my co-supervisor Dr. Rachel McDonnell from Trinity College Dublin for her support and guidance. I would like to thank Dr. Gabriel Costache and Sathish Mangapuram for their help and guidance during my internship at FotoNation. Thanks to my other FotoNation colleagues Dr. Joseph Lemley, and Dr. Barry McCullagh for their guidance and help. My special thanks to Dr. Claudia Costache for all of her help and administrative support.

I would like to thank my Ph.D. group friends and colleagues Viktor Varkarakis, Muhammad Ali Farooq, Waseem Shariff, Wang Yao, Dan Bigioi, Ayush K. Rai, and Sam Duignan. A special thanks to Faisal Khan for helping me with my experiments and other personal support during these four years. I want to express my thanks to Dr. Hossein Javidnia for guiding me to set up my initial research experiments.

Thanks to my housemates Sudeshna and Mohan for their support. A special thanks to my Bengali friends in Galway Kaustava, Ratul, Rajib, and Kaushik for those wonderful evenings we spent together venting our frustrations and discussing the never-ending Bengal politics. I

express my warm gratitude to my elder brother in Galway Rajarshi Hazra and his wife Tina Hazra for constantly encouraging me and making my life easier in Galway with all of their help and support.

Finally, my heartwarming gratitude, thanks, and respect go to my parents back in India without whom I will not be in Ireland to pursue my dream. Their constant encouragement and lifelong sacrifice make me what I am today. Last but not least I would like to thank my loving wife Shibani who believed in me and constantly motivated me to achieve my goal.

I would like to acknowledge Science Foundation Ireland and FotoNation for their generous funding and support. I want to thank the staff, co-workers, and my GRC committee members at the School of Computing of the University of Galway for all of their help and support.

Funding source: This work was conducted with the financial support of the Science Foundation Ireland Center for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224.



OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Abstract

Facial analysis tasks are of pivotal importance in social interaction, thus gaining extensive attention among the scientific community. With the increased popularity of deep learning models and the availability of high-performance infrastructure, it has become the de-facto tool for different facial analysis applications. However, when it comes to 3D facial analysis tasks like 3D face alignment, face reconstruction, facial expression analysis, etc., the availability of high-quality 3D face data is the biggest bottleneck. Particularly collecting accurate real ground truth pose and depth information is very challenging because of the limitations of real-world sensors. Furthermore, with the recent introduction of data privacy laws like GDPR and their associated restrictions, collecting face datasets has become more challenging, as it involves human subjects. With the advancement of computer graphics tools, domain-specific data generation with accurate annotations has provided a feasible alternative to real data. Though synthetic data can be a choice for deep learning training, the resulting domain gap between synthetic and real environments is still a challenge for the trained model to perform well in a real-world scenario. As a result, another type of approach has gained popularity: unsupervised learning, where the model tries to learn the objective without any annotated data.

In this dissertation, we address the issue of the unavailability of high-quality, accurate real-face data by applying these two approaches. With the help of low-cost digital asset creation software and an open-source computer graphics tool, we first build a pipeline to create a large synthetic face dataset. We rendered around 300k synthetic face images with extensive data diversity, such as different scene illuminations, backgrounds, facial expressions, etc., with their ground truth annotations like the 3D head pose and facial raw depth. We validate the synthetic data with two different facial analysis tasks - head pose estimation and face depth estimation. While learning the head pose from the synthetic images, we propose an unsupervised domain adversarial learning methodology to reduce the domain gap between the real and synthetic face images. We show that using our method, we can achieve near-state-of-the-art (SOTA) results with unsupervised training compared to the supervised methods that solely use real data to train their model.

Furthermore, to solve the scarcity of 3D face data, we propose a weakly supervised approach to extract the 3D face information from a single 2D face image. For this 3D face reconstruction task, we use the popular vision transformer with hierarchical feature fusion as the feature extractor module and train our network with a differential renderer in an unsupervised fashion without any real 3D face scan data. Though this approach is able to generate accurate 3D face shape from a single 2D face image, the model size is large and requires high computational resources. This makes it unsuitable for low-cost consumer electronic devices or processing at the edge. So in the last section of this thesis, we propose a pipeline to build 3D facial dense landmarks with 520 key points that cover the entire face as well as carry the information of the overall facial structure. To show that the data generated by our proposed method is able to preserve the 3D information, we train a dense face landmark predictor with this data. The trained model achieves comparable results to other SOTA methods in the sparse 3D facial alignment task.

Table of contents

List of figures	xiii
List of tables	xv
Nomenclature	xvii
1 Introduction	1
1.1 Introduction	1
1.2 Summary of Contributions	5
1.2.1 Synthetic Data Generation for Facial Analysis	5
1.2.2 Validation of Synthetic Facial Data through Computer Vision Tasks	5
1.2.3 Human Face Reconstruction from a Single Image with weak Super- vision	6
1.2.4 Lightweight Dense Face Landmark Detection	7
1.2.5 Other Contributions	7
1.3 List of Publications	8
1.4 Contribution Taxonomy	9
2 Synthetic Facial Data Generation	11
2.1 Background	11
2.2 Research Objective	13
2.3 Summary of Contributions	13
2.3.1 Generation of virtual Human Models	14
2.3.2 Setup of virtual Scenes	14
2.3.3 Collecting facial Ground Truths	15
2.4 Discussion on Contribution	16
2.4.1 Synthetic Head Pose Data	17
2.4.2 Synthetic Facial Depth Data	18
2.5 Copy of Published Works	20

3	Head Pose and Facial Depth Estimation using synthetic Facial Data	43
3.1	Background	43
3.2	Research Objective	46
3.3	Summary of Contribution	46
3.3.1	Learning Head Pose from synthetic Data through Regression	46
3.3.2	Learning Head Pose from synthetic Data through adversarial Domain Adaptation	47
3.3.3	Monocular Facial Depth Estimation from synthetic Images	48
3.4	Discussion on Contribution	50
3.5	Copy of Published Works	52
4	Face Reconstruction with weak Supervision	101
4.1	Background	101
4.2	Research Objective	102
4.3	Summary of Contribution	103
4.4	Discussion on Contribution	104
4.5	Copy of Published Works	108
5	Lightweight dense facial Landmark Prediction	125
5.1	Background	125
5.2	Research Objective	126
5.3	Summary of Contribution	127
5.3.1	Dense Facial Landmark Data Generation from UV Map	127
5.3.2	Dense Facial Landmark Prediction using Regression	129
5.4	Discussion on Contribution	134
6	Additional Contributions	137
6.1	Speech-driven Video Editing via Audio-Conditioned Diffusion Model	137
6.1.1	Background	137
6.1.2	Research Objective	139
6.1.3	Summary of Contribution	140
6.1.4	Discussion on Contribution	141
6.2	A Review of Benchmark Datasets and Training Loss Functions in Neural Depth Estimation	141
6.3	Copy of Published Works	142

7	Conclusion and Future Work	179
7.1	Contribution to the Generation of synthetic Face Data	179
7.2	Contribution to the Validation of the generated synthetic Face Data	180
7.3	Contribution to the unsupervised Face Reconstruction from a single Image .	181
7.4	Contribution to the Data Generation and Model Building for dense Face Landmark Estimation	182
	References	183

List of figures

2.1	Virtual Human Models from iClone : (a) Applying different model textures and shapes to a base model (b) Changing facial morph to add variations (c) Applying facial expressions on the facial models	15
2.2	Samples from our dataset: first row and second row are the RGB and depth pair with a simple background, the third and fourth rows are the RGB and depth pair with a textured background, and the fifth and sixth row are the RGB and depth pair with complex scene background.	16
2.3	Samples of fully rigged male and female models in fbx format imported in Blender.	17
2.4	Distribution of Head Pose Data (Yaw, Pitch, and Roll): The top row shows the distribution from our synthetic dataset, and the bottom row shows a similar distribution from the Biwi dataset	18
2.5	Sample depth data visualised from BIWI (face cropped) [47], Eurocom Kinect [98], Pandora (face cropped) [21] and our dataset.	19
3.1	Applying domain adaptation to train a cross-domain classifier [123]	44
3.2	Generic training framework for unsupervised domain adaptation introduced by Ganin and Lempitsky [52]	45
4.1	Qualitative comparison of the generated face shape with previous works - GANFIT [57], Tewari et al. [129], Tran & Liu [138]. The results of the previous works are taken from GANFIT [57]	104
4.2	Qualitative comparison of shape and texture on occluded images as ground truth with previous work - Tiwari et al. [133], Deng et al. [40], MOFA [130]. The results of the previous works are taken from Tiwari et al. [133]	105

5.1	UV position map example from [49]. Left: 3D plot of the corresponding position map on top of the 2D RGB image. Right: The first row is the 2D RGB image and the corresponding extracted texture and position map. The second row shows the x,y, and z channels of the position map data.	127
5.2	Selection of key points through Delaunay Triangulation. (a) Initial selected key points across the jaw, forehead, and nose tip (b) First iteration of Delaunay triangulation and centroid selection (c) Second iteration of Delaunay triangulation and centroid selection (d) Third iteration of Delaunay triangulation and centroid selection	128
5.3	(a) Template mesh in Blender (b) Final selected vertices highlighted on the template in Blender	129
5.4	Sample ground truth data. 1st column in the RGB image, 2nd column shows the selected 68 key points, 3rd column shows the fully selected 43867 vertices. 4th column shows the final selected 520 vertices.	130
5.5	Loss function comparison: L2, L1, smoothL1, Wing (with $w = 15$, $\epsilon = 3$). The plot shows the loss value against the error between the ground truth and the predicted value [83]. The quadratic growth of the L2 loss makes it sensitive to the outliers, while L2, L1, and smoothL1 yield a very small value for small errors between the ground truth and the predicted values. On the contrary, Wing loss is less sensitive to outliers and is much more sensitive to medium-to-small errors, which improves the training overall.	132
5.6	Cumulative Errors (NME) Distribution (CED) curves on AFLW2000-3D. Evaluation is performed on 68 landmarks with coordinates. Overall 2000 images from the AFLW2000-3D dataset are used. The backbone and loss functions are also shown in the legend. WL stands for Wing Loss, and L2 stands for MSE loss	134
5.7	Cumulative Errors (NME) Distribution (CED) curves on AFLW with 21 point landmarks. Overall 21k images from the AFLW dataset are used here. The backbone and loss functions are also shown in the legend. WL stands for Wing Loss, and L2 stands for MSE loss.	135

List of tables

2.1	Author's Contribution to [14, 15]	14
3.1	Author's Contribution to [16]	47
3.2	Author's Contribution to [12]	48
3.3	Author's Contribution to [84, 86]	49
4.1	Author's Contribution to [13]	103
4.2	Subjective evaluation results in four different hypotheses - Realism, texture, shape reconstruction, occlusion resistance. The table shows the Mean Opinion Score (MOS) and the standard deviation (Std.)	106
5.1	Quantitative evaluation on AFLW2000-3D dataset on facial alignment task.	133
5.2	Quantitative evaluation on AFLW dataset with 21-point landmark definition on facial alignment task.	133
5.3	Comparative analysis with two different backbones Mobilenet-V2 and Resnet-18 of Quantitative result on AFLW-3D dataset on facial alignment task and the computational requirement.	133
6.1	Author's Contribution to [19]	140

Nomenclature

Acronyms / Abbreviations

2D 2 Dimensional

3D 3 Dimensional

3DMM 3D Morphable Model

BFM Basel Face Model

CED Cumulative Errors Distribution

CG Computer Graphics

C – GAN conditional Generative Adversarial Network

CNN Convolution Neural Network

DA Domain Adaptation

DNN Deep Neural Network

FAM Feature Aggregation Module

fbx Filmbox

FOV Field Of View

FR Face Recognition

GAN Generative Adversarial Network

GDPR General Data Protection Regulation

GPL General Public License

HMM Hidden Markov Model

HPE Head Pose Estimation

IMU Inertial Measurement Unit

LSTM Long Short-Term Memory

MOS Mean Opinion Score

NME Normalized Mean Error

PCA Principle Component Analysis

SOTA State Of The Art

SSIM Structural Similarity Index Measure

ViT Vision Transformer

Chapter 1

Introduction

1.1 Introduction

With the world population ever increasing, there is no doubt that with more than 7 billion unique samples, human faces are one of the most complex data types in computer vision. Though the base structure of human faces is similar to each other, the detailed characteristics and deformation vary significantly with the variation of age, ethnicity, and gender. Thus human faces are always a prevalent subject among computer vision researchers in different tasks such as face detection [91], identity [145] and expression [94] recognition, face re-enactments/swapping [42], random face generation [164, 144], face modeling or reconstruction [99, 170] etc. These facial analysis tasks are extensively exploited in different applications, which include security [125, 119, 72, 74], human-computer interaction [33, 124], animation [46, 149, 37] and even health [105, 154]. With the advancement of deep neural networks (DNN), it is now possible to produce human-level performance in different computer vision tasks, which makes it the obvious choice for facial analysis tasks too [170, 146, 145, 94, 164]. Though the performance of these DNN models largely depends on the massive amount of accurate ground truth training data. Due to the availability of large-scale 2D face data, 2D facial analysis has been used widely for many years. But because of the 3D nature of the human face, 2D images fail to accurately capture the complex geometry, as it collapses into one dimension. Also, 3D imaging comes with a geometrical representation invariant to pose and scene illumination, a significant drawback of 2D imaging.

These recent advancements have made 3D deep learning popular among researchers. But it comes with its own price of the scarcity of the 3D ground truth data, which often limits its scope. 3D facial data can be produced by 3D scanners, stereo-vision systems, or RGB-D sensors (e.g., Microsoft Kinect). The first two methods can acquire high-quality 3D face data

but require a controlled environment and expensive equipment. On the other hand, RGB-D cameras are comparatively cheaper, but the captured data is of limited quality. Overall all these methods have a significant issue of not covering all the variations across ethnicities and age groups, as the data is mainly acquired in a controlled setup. Thus machine learning models trained with these data fail to generalize enough to work in real-world industrial use cases and limit the fairness of the model [60]. So instead of collecting and labeling real data, which is an expensive task that can be subject to bias, an alternative solution, the synthesizing of training data using computer graphics (CG) tools, has been introduced. With synthetic data, we have complete control over the variations of the data, thus eliminating biases. This also ensures perfect labels without any annotation noise, which is otherwise impossible to label by hand. The computer vision community has studied synthetic data in different tasks like scene understanding [114, 53, 36], eye tracking [150, 117, 22], hand tracking [162, 141], object recognition [75], full body analysis [78, 73] and many others [102]. Though advancements in generative deep learning models, GAN and diffusion models can create high-quality and realistic 2D face data. However, very few previous works have attempted to synthesize a full 3D human face due to the human face's complexity. So there is a significant gap in the current research on creating and utilizing synthetic 3D face databases using the available CG toolchains.

With CG tools, we can generate pixel-perfect synthetic data for most computer vision tasks that can be used to train deep learning models. But when it comes to the problem of high-level computer vision tasks like object detection, object or scene segmentation, 3D pose, viewpoint- and depth estimation, research has shown that the domain gap between the synthetic and real data does not allow for achieving SOTA results by training only on synthetic data. So researchers have tested hybrid datasets, a mix of real and synthetic data, and achieved better results. Also, Movshovitz-Attias et al. [100], and Tsirikoglou et al. [136] showed that making the synthetic data more realistic with advanced CG tools helped to improve the results in tasks like viewpoint estimation and object detection. Particularly when it comes to the problem of human facial analysis, photo-realism of synthetic data is a major issue. This domain gap (or domain shift) between the real and synthetic data can be eliminated by domain adaptation (DA) techniques. DA is a subcategory of transfer learning where we try to make the model trained on one domain of data which we called the source domain, so that the trained model will perform well on a different domain or the target domain. Unlike other transfer learning methods here the feature space of both the domains remains the same but the distribution of the data differs from each other. Synthetic 3D face data can be a good candidate for DA as we can try to train the models with synthetic data and

evaluate the performance against real face data. However, such direct transfer of knowledge may not work well due to domain shift or dataset bias. Fine-tuning the pre-trained source model with a small sample of the target labeled data can be a solution. But fine-tuning still requires a considerable amount of target domain data, which is not available when it comes to 3D facial analysis tasks. Additionally, almost all these domain adaptation methods are studied for classification tasks where some classes do not exist in the target domain. But most facial analysis tasks like head pose estimation or learning the 3D face data from the depth queues fall under regression problems where the DA is not yet studied extensively.

The use of synthetic data for 3D facial analysis can solve some of the data scarcity issues. However, still, it fails to replicate the distributions of the intrinsic characteristics of real faces. On the other hand, capturing high-volume 3D scans is expensive and such datasets are unavailable for model training. So an easy and feasible alternative to capturing a 3D scan of a face is to estimate the face geometry from uncalibrated 2D face images. But due to the complex nature of the human face, this approach of 3D-from-2D reconstruction is inherently ill-posed, as we need to recover the facial geometry, head pose, and texture information (including the color and illumination) from a single 2D face image. Also, a single 2D picture can be generated from the different 3D models as long as the texture matches the 2D image, so it generates ambiguities in these 2D-to-3D solutions. A well-agreeable solution is to add prior knowledge to resolve these ambiguities. As the human face has a common base shape, this can be used as prior knowledge for any face reconstruction task. So statistical 3D face models are the most popular way to add this prior knowledge, as they have the ability to encode geometric variations with appearance properties. The most commonly used statistical face model is the 3D Morphable Model (3DMM) proposed by Blanz and Vetter [20], which consists of the shape (geometry) and the albedo (texture or color) model constructed from a set of high-quality 3D face scans using Principle Component Analysis (PCA). But in order to train the model with 3DMM data, we need a set of ground truth images and their corresponding 3DMM parameters, which is often not available.

So a new strategy has become popular: self-supervised training - at first, the 3DMM parameters are predicted through a backbone network, then a 3D face model is built with the help of those predicted 3DMM parameters and fed to a differential renderer layer that renders the predicted 3D face model to the image plane. Finally, the rendered 2D image is compared with the ground truth image. Most of these self-supervised networks trained a CNN backbone to learn the 3DMM parameters. By its fundamental characteristics, convolutions are local operations. Thus sometimes Convolution Neural Networks (CNN) fail to learn the global

features, which is essential when it comes to face reconstruction tasks. To overcome this shortcoming of CNN in the computer vision community, transformer-based architectures have gained immense popularity due to their ability to capture long-term dependencies. Though these vision transformers (ViTs) have achieved SOTA results in different computer vision tasks like image classification [143, 97, 44, 147], object detection [160, 48], and image segmentation [163, 43, 62], transformers are not studied in face reconstruction tasks. This opens up a new area where the effectiveness of different vision transformer networks can be studied to face reconstruction problems.

In the analysis-by-synthesis method, which is discussed above, the model is learned by reducing the photometric error [65] between a generative 3D face model and a ground truth image using differentiable rendering techniques. But to make this differentiable rendering computationally feasible, it depends on a number of approximations. It assumes the human face as a Lambertian object, and the reflectance model and the scene illumination as spherical harmonics alone [151]. But in reality, the complexity of the human face can not be modeled as a linear Lambertian object. Also, the illumination effects, such as shadows cast by the nose or the ambient occlusion, can not be modeled by spherical harmonics. To alleviate these limitations, either we have to rely on the fit-and-render strategy, where the face is fitted to a 3DMM, or we need to train a more complex and large deep learning model. Both these approaches make the model computationally expensive and inappropriate for edge and IoT devices. An alternative to this is extracting the facial landmarks, which are the points of correspondence across the faces. Almost all of the publicly available facial landmark datasets have only 68 key points. But the overall facial expressions and identities can not be encoded by only 68 sparse landmarks. When it comes to reconstructing the whole face, it is almost impossible to get relevant information from those landmarks. But if we are able to predict dense landmarks which cover the entire face, it will help to get the face shape. It will also help to eliminate the dependence on statistical models like 3DMM and make the model size reasonable to make it work in edge devices.

The major challenges in the 3D facial analysis that are identified and addressed in this thesis are therefore: 1) lack of high-quality 3D face data with accurate ground truth annotations like the head pose and face depth, 2) feasibility of synthetic face data as an alternative of real data in popular computer vision tasks like head pose estimation and face depth estimation, 3) photorealistic 3D face synthesis without ground truth 3D face scans, 4) limitations of the unsupervised or semi-supervised face reconstruction model due to its computational complexity which limits them to run on the edge devices.

1.2 Summary of Contributions

In this section, a short summary of the main contributions is presented. In the later chapters, each of these contributions is discussed in more detail. Each chapter will start with an introduction which will present the context of the research, followed by the motivation against that section. This will lead to the main research question addressed in that chapter. Each chapter will end with a contribution to that research question and discussion. Additionally, for each work, a table is presented showing the contribution of the authors in that article.

1.2.1 Synthetic Data Generation for Facial Analysis

Chapter 2 contributes to a methodology for generating synthetic facial data. In the initial research work [14], a methodology for building synthetic face data is presented. With the help of commercially available synthetic asset-building software and an open-source CG tool, a pipeline to build synthetic face data with their corresponding annotations, like the head pose and face depth, is proposed. With the help of the proposed pipeline, a dataset has been constructed and released for public use in the subsequent work [15]. The published dataset has two sets of data consisting of ground truth head pose and depth map. The head-pose dataset has more than 600k pairs of synthetic face images and their corresponding ground truth head-pose annotations. The facial depth data set has more than 500k of synthetic face data with their raw depth data.

1.2.2 Validation of Synthetic Facial Data through Computer Vision Tasks

The synthetic data that was generated, as mentioned in Chapter 2, is then validated through two different computer vision tasks. These methods of validating head pose and facial depth are discussed in Chapter 3. In the first part of the study, the task of measuring the accurate head pose from a single headshot image is considered. Though there are many studies on head pose estimation using popular real datasets, these methods are highly biased on the limited data available. There is minimal work on learning head pose from a synthetic dataset. So in the initial work [16], a SOTA model has been trained solely with the new proposed synthetic data. Near SOTA head pose estimation (HPE) results are achieved in compared to the SOTA models which are trained on only real data. Also, a data-fusion-based transfer learning approach is applied, where the model trained with the synthetic data is fine-tuned with only 1k of real data. The result of the model surpasses the current SOTA results. This initial work is further expanded by proposing an adversarial DA approach [12]. In this work,

the model is trained simultaneously on the labeled synthetic data in supervised and unlabeled real data in an unsupervised way. The model achieved significantly better results than the model trained on synthetic data only.

In the next study, the validity of the raw depth data is studied through the facial depth estimation task. An initial work [85, 84] is first presented on learning an accurate face depth estimation model. A shallow autoencoder-based deep learning model is trained with the synthetic face data and their corresponding ground truth raw depth data. The initial experiments show promising results when the model is evaluated against the synthetic test and evaluation dataset; a simple, less complex model does provide better results than the dense feature extractor models. This work is further extended in [86] where a hybrid loss function is proposed to learn the accurate depth from a single image training a light-weight encoder-decoder based depth estimation model. A detailed ablation study is also conducted, varying different backbones of the encoder network and changing the weights of the loss terms to see the individual contribution of the different loss terms. Through multiple experiments, it has been found that the proposed lightweight model is more computationally efficient than the current SOTA depth estimation models and shows a performance equal to or better than the SOTA when evaluated across four different public datasets.

1.2.3 Human Face Reconstruction from a Single Image with weak Supervision

In chapter 3, we achieved a near SOTA result in learning facial depth training on synthetic data. But still, due to a lack of real data, we are not able to validate the learned model extensively. So in the next work, as presented in chapter 4, we propose a weakly supervised learning framework for 3D face reconstruction from a single facial image. For 3D face reconstruction, a statistical face model like 3DMM acts as a powerful prior. Recent works proposed several methods that build on top of predicting the 3DMM parameters to form the face mesh. When it comes to predicting a 3D face from a single face image, we need to extract the features from that image. CNN has gained popularity as the de-facto feature extractor for most computer vision tasks. It is efficient in learning local patterns. But it fails to capture the long-range dependencies between patches, which is essential when it comes to the face reconstruction task. So recently, transformer-based networks have been adopted by the computer vision community for their ability to learn long-range dependencies. But at the same time, these lack the ability to learn the local features compared to CNN. So in this work, we propose a hierarchical feature aggregation module-based feature extractor with the Swin Transformer as its backbone as the feature extractor. This architecture is able to

learn multi-scale features in a coarse-to-fine manner with a reduced computational cost and model size compared to vanilla transformers. To the best of our knowledge, this is the first work that has studied the effectiveness of transformer networks with feature fusion in the face reconstruction task. With the proposed architecture, we train the network in a weekly supervised manner. We predict the 3DMM parameters from the single-face image, pass them through a differential renderer, and compare the rendered image with the ground truth image without any ground truth 3D face scans or facial depth cues.

1.2.4 Lightweight Dense Face Landmark Detection

Chapter 5 presents a new method for learning dense 3D landmarks from a monocular face image. At first, a ground truth of dense face landmarks of 520 key points is obtained from the UV map data that was originally developed in [49]. We then train a lightweight regressor network to learn the key points from those ground truths. We have conducted a detailed ablation study on the model performance and varying computational complexities.

1.2.5 Other Contributions

In chapter 6, we presented two of the secondary publications. The first section provided the details about the work on Speech Driven Video Editing via an Audio-Conditioned Diffusion Model. Facial video editing with audio cues is a very popular and complex facial analysis task. The goal of this task is to re-synchronize the lip and jaw movement of a speaker in a video based on a new speech input signal. Here, we presented an end-to-end method for speech-driven video editing with a diffusion-based generative model. Though facial landmarks or other facial reconstruction queue help as intermediate learning, we have not relied on them because of a lack of ground truth. Instead, we propose an unstructured generation method that directly generates the facial video conditioned by the audio signal. We have used a U-Net-based denoising diffusion model based on Palette [118]. Our method impaints the lower half region of the face, including the lip and jaw movements. We conditioned the network with mel-spectrogram features combined with the previously generated frames to generate the next frame. This helps to add the audio signal as well as maintains temporal stability.

In the next section, a work based on a review is presented. During the previous study of facial depth estimation using synthetic data, we also reviewed the different loss functions used as an objective for the depth estimation task and different datasets used for training. The detailed review is published in this work [87].

1.3 List of Publications

Synthetic Data Generation for Facial Analysis

1. Basak, Shubhajit, Hossein Javidnia, Faisal Khan, Rachel McDonnell, and Michael Schukat. "Methodology for building synthetic datasets with virtual humans." In 2020 31st Irish Signals and Systems Conference (ISSC), pp. 1-6. IEEE, 2020.
2. Basak, Shubhajit, Faisal Khan, Hossein Javidnia, Peter Corcoran, Rachel McDonnell, and Michael Schukat. "C3I-SynFace: A synthetic head pose and facial depth dataset using seed virtual human models." *Data in Brief* (2023): 109087.

Validation of Synthetic Facial Data through Computer Vision Tasks

3. Basak, Shubhajit, Peter Corcoran, Faisal Khan, Rachel McDonnell, and Michael Schukat. "Learning 3D head pose from synthetic data: A semi-supervised approach." *IEEE Access* 9 (2021): 37557-37573.
4. Basak, Shubhajit, Faisal Khan, Rachel McDonnell, and Michael Schukat. "Learning accurate head pose for consumer technology from 3D synthetic data." In 2021 IEEE International Conference on Consumer Electronics (ICCE), pp. 1-6. IEEE, 2021.
5. Khan, Faisal, Shubhajit Basak, Hossein Javidnia, Michael Schukat, and Peter Corcoran. "High-Accuracy Facial Depth Models derived from 3D Synthetic Data." In 2020 31st Irish Signals and Systems Conference (ISSC), pp. 1-5. IEEE, 2020.
6. Khan, Faisal, Shahid Hussain, Shubhajit Basak, Joseph Lemley, and Peter Corcoran. "An efficient encoder–decoder model for portrait depth estimation from single images trained on pixel-accurate synthetic data." *Neural Networks* 142 (2021): 479-491.

Human Face Reconstruction from a single Image with weak Supervision

7. Basak, Shubhajit, Peter Corcoran, Rachel McDonnell, and Michael Schukat. "3D face-model reconstruction from a single image: A feature aggregation approach using hierarchical transformer with weak supervision." *Neural Networks* 156 (2022): 108-122.

Other Contributions

8. Bigioi, Dan, Shubhajit Basak, Michał Stypułkowski, Maciej Zieba, Hugh Jordan, Rachel McDonnell, and Peter Corcoran. "Speech driven video editing via an audio-conditioned diffusion model." *Image and Vision Computing* (2024): 104911.

9. Khan, Faisal, Shahid Hussain, Shubhajit Basak, Mohamed Moustafa, and Peter Corcoran. "A Review of Benchmark Datasets and Training Loss Functions in Neural Depth Estimation." *IEEE Access* 9 (2021): 148479-148503.

1.4 Contribution Taxonomy

As this thesis is an article-based submission, the works included have been done with the collaboration of multiple authors. In order to establish the primary authorship of the listed papers, the CRediT [5] methodology has been adapted. CRediT is a popular taxonomy followed by most of the reputed journals to specify the contribution of the authors. It is measured based on 14 roles: Conceptualization, Data curation, Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Though the works are done in collaboration, the majority of the work presented in this thesis is done by me. In this thesis, the taxonomy is more simplified, and the contributions are broadly categorized in the following -

- Ideation - This includes conceptualization and ideation of the main hypothesis.
- Experiments & Implementations - This includes methodology, investigation, data curation, software development, validation, and visualization.
- Manuscript Preparation - This includes all aspects of writing the drafts, including Writing – the original draft, Writing – review & editing.
- Background Work - This includes some aspects of literature review, formal analysis, resourcing, project administration, and supervision to ensure that the methodology used is typical of that used in the field publication area.

This simplification of CRediT ignores most aspects of project funding, project administration, and overall supervision but otherwise encapsulates the main attributes of the primary authorship as per CRediT. Each of the consecutive chapters is accompanied by a table in the 'Summary of Contribution' section showing the contributions of each author to the aforementioned four criteria. Authors are listed by their initials where SB stands for Shubhajit Basak, FK stands for Faisal Khan, HJ stands for Hossein Javidnia, PC stands for Peter Corcoran, RM stands for Rachel McDonnell, MS stands for Michael Schukat, JL stands for Joseph Lemley, DB stands for Dan Bigoi, and SH stands for Shahid Hussain. Contributions are presented as a percentage (%) of work that falls under these four categories.

Chapter 2

Synthetic Facial Data Generation

2.1 Background

Recent advancements in CGI technology have improved the quality of synthetic data and made it popular for deep learning training [103]. Particularly in low-level computer vision tasks such as optical flow estimation (estimating the distribution of the apparent movement of different objects, edges, and surfaces caused by the relative motion of the observer with respect to the scene) or stereo image matching (finding correspondence between two points in the same image captured from two different viewpoints), etc., synthetic data has been used extensively, as these tasks can be approached with methods that do not require large real data repositories or much learning in terms of deep learning methods. One of the recent works by Mayer et al. [96] provides an overview of different synthetic datasets for low-level computer vision tasks specifically for optical flow. Through extensive experimental results, they show that the realism of the synthetic data is not a significant requirement for low-level tasks explained above. Instead, combining the different synthetic datasets, which vary in situations and domains, and adding real-life simulations like lens distortion, image blur, or Bayer interpolation artifacts in the synthetic dataset, improves the result of the models significantly. When it comes to high-level tasks like object detection or segmentation, the quality and realism of the synthetic data play a significant role in model training and performance.

Particularly, synthetic models of the human face and human body are of immense interest among the scientific community, as they have advantages over real-face datasets, which have some major issues like:

- The real face datasets often contain biases regarding gender, race, and other parameters [90, 80].

- Labeling of face attributes like head pose, eye gaze, facial key points, or, more importantly, face shape is expensive and hard to achieve manually.
- Privacy and ethical issues and regulations like GDPR often restrict the usage of real-face datasets [109].

In contrast, synthetic facial data comes with its own perks:

- With the help of CGI tools, we can generate a theoretically unlimited amount of face data with control over different properties like head pose, eye gaze, gestures, etc. At the same time, we can generate accurate ground truth labels like face segmentation, facial key points, and joint locations, which is almost impossible to gather for real data.
- Synthetic data can be used to augment real datasets, reducing the bias of real datasets. Generative models with domain adaptation approaches can make these synthetic data more realistic.

In one of the earliest works on synthetic faces, Queiroz et al. [107] proposed a pipeline to generate the ground truth of real faces with realistic textures extracted from real faces and published the Virtual Human Faces Database (VHuF). Later Bak et al. [9] published the Synthetic Data for person Re-Identification (SyRI) dataset generated with the help of Adobe Fuse CC and Unreal Engine 4. They created the scene lighting based on HDR environment maps. Hu et al. [71] proposed a pipeline to generate synthetic face datasets by combining automatically detected facial parts like eyes, nose, mouth, etc., and used the data for face recognition (FR) tasks. Their results showed that the resulting artifacts in the faces did not affect the FR accuracy and, in some cases, improved the robustness of the model. Few other recent works used 3DMM models to generate some parts of the face and used them in specific tasks. For example, both Sugano et al. [127] and Wood et al. [153] used 3DMM-based eye models for gaze estimation tasks. But due to the complexity of the full human face, very few previous works have attempted to generate full-face synthesis with computer graphics pipelines. The most recent and relevant work was published by Wood et al. [150], where they generated a large face dataset through a pipeline by combining a parametric face model with a large set of high-quality artists created CG assets like textures, hair, and clothing. Though the dataset has annotations like dense landmarks, normal maps, depth, and face segmentation, only the sparse landmarks and segmentation are made publicly available. None of these datasets contains the 3D models and depth cues, which are the main attributes of 3D face analysis.

2.2 Research Objective

As stated in the previous section, there is a very limited amount of open-source synthetic face data currently available online. Particularly when it comes to learning 3D cues, no such dataset has 3D annotations. So our main objective of this study is to build a large synthetic face dataset with ground truth annotations that can be used to learn 3D face shapes. With the advancement of CG technology, currently, there are many open-source CG tools (like Blender, Krita, Lunacy, Gimp, 3D-Max etc.) that are publicly available. With the help of these software chains, we can build a pipeline to create a large-scale synthetic face dataset and collect ground truth annotations. Though to build the face dataset, we need virtual human models. A common option can be collecting 3D face scans. But they are expensive and involve setting up complex environments. So to achieve our goal, we perform the following:

- Search through the available CG tools (like Maya, 3DS Max, Blender, Cinema 4D etc.) and select the appropriate and useful one in terms of usability and ease of learning.
- Identify the 3D synthetic human models that are available in the online market and cheap to buy and use.
- Provide enough variations in facial expressions and gestures as well as appearances.
- Build an automated pipeline using a programming language like python to generate the ground truth face images with their annotations like the head pose and facial depth. Also, to build the ability to provide control over head movement and background scenes within the pipeline.
- Finally, using the pipeline, build a large dataset that can be used for deep learning training.

2.3 Summary of Contributions

This main work is presented through the article - Basak, Shubhajit, Hossein Javidnia, Faisal Khan, Rachel McDonnell, and Michael Schukat. "Methodology for building synthetic datasets with virtual humans." at the 2020 31st Irish Signals and Systems Conference (ISSC) [14]. The resulting dataset is presented in - Basak, Shubhajit, Faisal Khan, Hossein Javidnia, Peter Corcoran, Rachel McDonnell, and Michael Schukat. "C3I-SynFace: A synthetic head pose and facial depth dataset using seed virtual human models." Data in Brief (2023): 109087 [15]. A copy of the published papers are attached at the end of this chapter.

Table 2.1 Author’s Contribution to [14, 15]

Contribution Criteria	Contribution Percentage
Ideation	SB 70%,HJ 20%,PC 10%
Experiments & Implementations	SB 90%,FK 10%
Manuscript Preparation	SB 90%,HJ 3%,FK 4%,RM 3%
Background Work	SB 70%,MS 20%,PC 10%

The contributions of the authors for the above-mentioned research work [14, 15] as per the four major criteria discussed in section 1.4 is presented in the table 2.1.

2.3.1 Generation of virtual Human Models

In order to achieve the research objective, the first step is to build the pipeline to create virtual human models. We have chosen a commercially available digital asset creation software called iClone 7 [3] and Character Creator [1] for creating the virtual models:

- Character Creator provides “Realistic Human 100” - which contains 100 virtual human models with variation over ethnicity, race, gender, and age.
- The morphing shape or the mesh of different parts of the body can be adjusted to give more variation over the shape.
- Additionally, different expressions like sad, angry, happy, scared, and neutral are added to the models.
- These models are then exported to fbx (Filmbox) format, which has the mesh and armature (bones) and can have facial expressions embedded as frames. So it can be used to exchange both geometry and animation data.

2.3.2 Setup of virtual Scenes

As we have these models, we need to import these to the CG software to put them into a scene and render them with ground truth annotations. We have chosen the open-source CG software Blender [2], as it is comparatively simple and has Python support to automate batch rendering. Also, Blender is released under the GNU General Public License (GPL, or “free software”), which allows us to use and distribute it freely. We put the models in three different scenes - 1. A scene with plain background with single color; 2. A scene with a textured plane background, where we have used the textures provided by Abdelmounaime

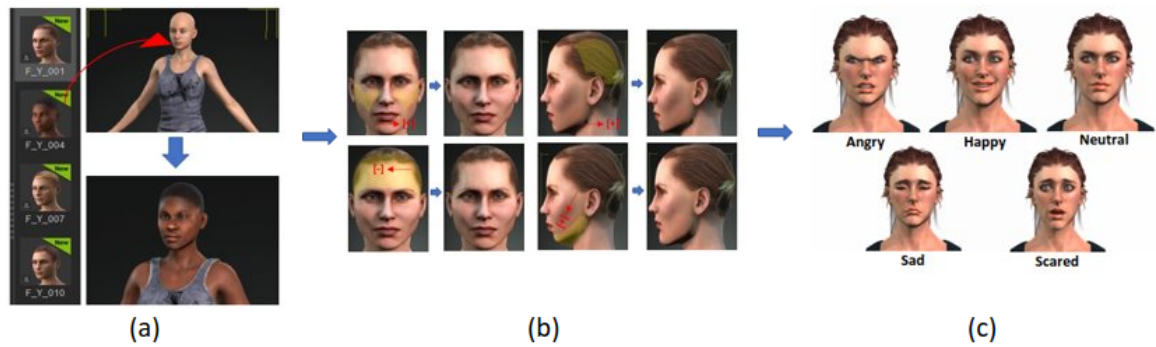


Fig. 2.1 Virtual Human Models from iClone : (a) Applying different model textures and shapes to a base model (b) Changing facial morph to add variations (c) Applying facial expressions on the facial models

and Dong-Chen [4]; 3. Two complex scenes (classroom and barbershop) were collected from the Blender marketplace. We have used the Cycles Rendering Engine in Blender to render the scene, as the Cycles engine offers ray-tracing capabilities for photo-realistic rendering. The whole process, from importing the models in Blender to setting up the scene, including adding scene illumination and camera and finally rendering the ground truth with the annotations, is automated by Python scripts. The code is made publicly available through a GitHub repository¹.

2.3.3 Collecting facial Ground Truths

The fbx models imported in Blender are scaled and put into different scenes. The Blender rendering camera field of view (FOV) and sensor size are set to 60 degrees and 36 millimeters, respectively. The ground truth is collected by setting up the RGB and Z-pass output in the Blender compositor layer for the RGB and raw depth data. Additionally, we also apply continuous rotations on the shoulder bone to vary the head pose. To cover all the cross-rotation angles similar to human head movements, we apply head rotations similar to the ground truth of the popular real head pose dataset BIWI [47]. We have published two separate datasets - one for the head pose, which contains around 300k ground truth RGB images and their corresponding head pose annotations, and one for facial depth data, which contains around 250k ground truth RGB images, their corresponding raw depth (in *.exr format) and

¹<https://github.com/shubhajitbasak/blenderDataGeneration>

head pose annotations. The fbx models, face depth data, and head pose data are publicly available in [link1](#)², [link2](#)³, and [link3](#)⁴, respectively.

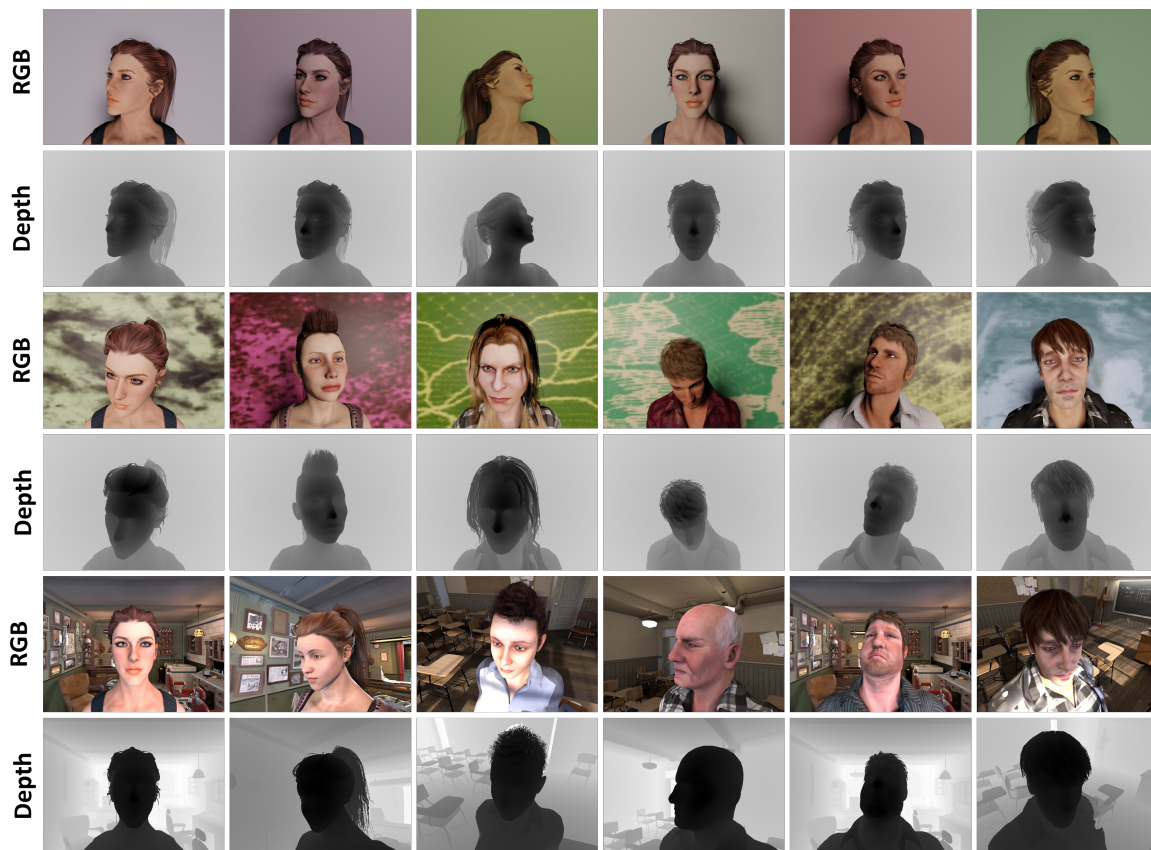


Fig. 2.2 Samples from our dataset: first row and second row are the RGB and depth pair with a simple background, the third and fourth rows are the RGB and depth pair with a textured background, and the fifth and sixth row are the RGB and depth pair with complex scene background.

2.4 Discussion on Contribution

This work provides a framework and an automatic pipeline to generate a large amount of synthetic facial ground truth data using low-cost virtual human models. As we have released the raw fbx synthetic models and the data generation scripts, one can generate a large amount

²https://drive.google.com/drive/folders/177Xem5rLg7GYRn6IDwWMwZtBgr57OrtB?usp=share_link

³https://drive.google.com/drive/folders/1oleqLbR793xBmw8gF91JTt4TrBJQUMr2?usp=share_link

⁴https://drive.google.com/drive/folders/10QNib4Rp9D7SHMbdIK3ecbZFIL_bNOEY?usp=share_link

of data varying the head pose, background scene, scene illumination, etc. Also, as these models are fully rigged, they can be animated using full-body mocap data to generate full-body animation data as well. Figure 2.3 shows an example of a male and a female fbx model with their armatures visible. Apart from the fbx models, we have also released the head pose and the face depth data. While the head pose dataset can be used in 3D face alignment tasks, the facial depth data can be helpful for 3D facial depth estimation and face reconstruction tasks. In the following subsections, we will discuss the uniqueness of the head pose and depth data in more detail.



Fig. 2.3 Samples of fully rigged male and female models in fbx format imported in Blender.

2.4.1 Synthetic Head Pose Data

A major challenge for learning-based head pose estimation methods is the requirement of accurately labeled data. Accumulating real head pose data for model training is difficult as it involves human subjects, which mostly raises ethical and data privacy issues. Also, data acquisition measurements like depth sensing or IMU motion are prone to sensor errors. The most popular head pose real datasets, like Biwi Kinect Head Pose Dataset [47] or Pointing'04 [61], only contain around 15k and 4k data samples collected from 20 and 14 subjects, respectively, which makes them not suitable for deep learning-based model training. The only large real dataset for head pose training available is 300W-LP [168], which is synthesized by fitting a 3D face model to the image and profiling the image to a large pose.

This dataset contains 61225 samples. Also, some of the datasets, like Bosphorus [120] and Pointing'04 [61], are discrete and only contain specific head pose angles. The only synthetic head poses dataset available is Synhead [63], which is rendered from high-quality face scans of 10 subjects. This makes this less diverse and expensive to acquire. On the contrary, the dataset produced by our work has more than 300k frames collected from 100 individual models, which makes it robust and suitable for deep-learning training. Also, as we have applied continuous rotations and applied the Biwi Head Pose sequence, it covers a wide range to head poses. Figure 2.4 shows the distribution of yaw, pitch, and roll of our generated data and the same distribution from the Biwi dataset.

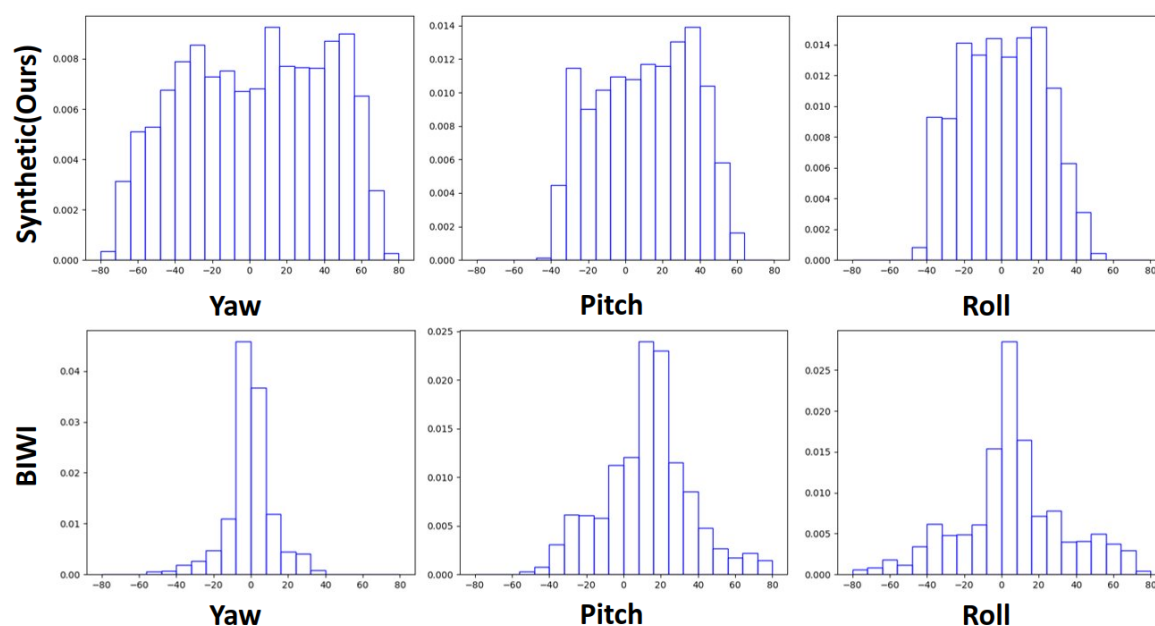


Fig. 2.4 Distribution of Head Pose Data (Yaw, Pitch, and Roll): The top row shows the distribution from our synthetic dataset, and the bottom row shows a similar distribution from the Biwi dataset

2.4.2 Synthetic Facial Depth Data

Collecting monocular depth information from scenes mostly relies on depth sensors like Kinect. But when it comes to facial depth data, it is very hard to acquire because of privacy and ethical issues. Also, the output of these depth sensors is not accurate as the depth output highly depends on the range and resolution of the sensor. For example, Dutta et al.[45] shows that if the Kinect is placed within the ideal range (1m to 3m) and with proper FOV, it is able to capture the 3D positions of a marker with very minimal error (< 1cm). Also, the depth data is prone to sensor noise and missing depth (or missing hole) issues. In contrast,

this work does not contain any of these issues, as we have collected the depth data with a CG toolchain. The most popular real-depth datasets that contain facial samples, such as Pandora [21], Biwi Kinect Head Pose [47], and Eurocom Kinect Face [98], have a limited sample size (250k, 15k, and 50k respectively) with fewer variations of subjects (24, 20, and 52 respectively). Also, it can be noted in particular that these datasets contain a very small amount of dynamic objects, as most of these datasets are acquired in a constrained environment with a plain background. So networks trained on these data with such a strong bias often fail to generalize properly. On the contrary, our data is rendered with three different backgrounds - plain, textured, and complex scenes with multiple objects, making it suitable for robust deep learning training. Figure 2.5 shows an example from the Pandora [21], Biwi Kinect Head Pose [47], and Eurocom Kinect Face [98] dataset. It can be observed all these datasets mostly have plain backgrounds and also contain noise in the depth data due to the limitation of the sensor.

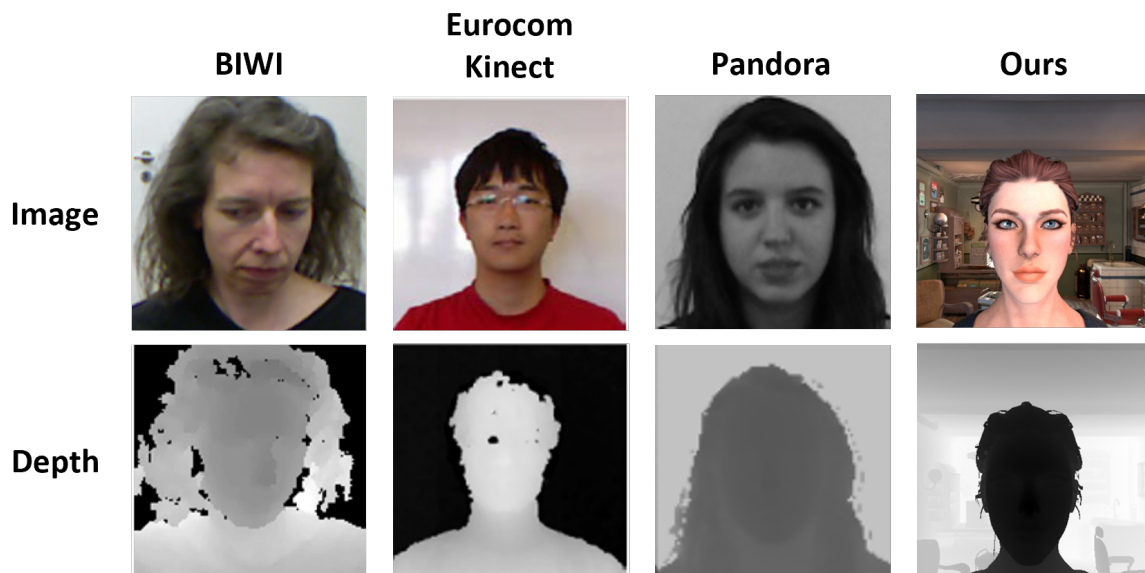


Fig. 2.5 Sample depth data visualised from BIWI (face cropped) [47], Eurocom Kinect [98], Pandora (face cropped) [21] and our dataset.

2.5 Copy of Published Works

Methodology for Building Synthetic Datasets with Virtual Humans

Shubhajit Basak
College of Engineering and Informatics
National University of Ireland, Galway
Galway, Ireland
s.basak1@nuigalway.ie

Hossein Javidnia
ADAPT Research Center
Trinity College Dublin
Dublin, Ireland
hossein.javidnia@adaptcenter.ie

Faisal Khan
College of Engineering and Informatics
National University of Ireland, Galway
Galway, Ireland
f.khan4@nuigalway.ie

Rachel McDonnell
School of Computer Science and
Statistics
Trinity College Dublin
Dublin, Ireland
ramcdonn@scss.tcd.ie

Michael Schukat
College of Engineering and Informatics
National University of Ireland, Galway
Galway, Ireland
michael.schukat@nuigalway.ie

Abstract— Recent advances in deep learning methods have increased the performance of face detection and recognition systems. The accuracy of these models relies on the range of variation provided in the training data. Creating a dataset that represents all variations of real-world faces is not feasible as the control over the quality of the data decreases with the size of the dataset. Repeatability of data is another challenge as it is not possible to exactly recreate ‘real-world’ acquisition conditions outside of the laboratory. In this work, we explore a framework to synthetically generate facial data to be used as part of a toolchain to generate very large facial datasets with a high degree of control over facial and environmental variations. Such large datasets can be used for improved, targeted training of deep neural networks. In particular, we make use of a 3D morphable face model for the rendering of multiple 2D images across a dataset of 100 synthetic identities, providing full control over image variations such as pose, illumination, and background.

Keywords— *Synthetic Face, Face Dataset, Face Animation, 3D Face.*

I. INTRODUCTION

One of the main problems in modern artificial intelligence (AI) is insufficient reference data, as in many cases available datasets are too small to train Deep Neural Network (DNN) models. In some cases, where such data has been captured without a label, the manual labeling task is time-consuming, costly, and subject to human error. Producing synthetic data can be an easier approach to solving this problem. For image data, this can be achieved via three dimensional (3D) modeling tools. This approach provides the advantage of extraction of the ground truth information from 3D Computer Graphics (CG) scenes. While this process still requires some manual labor to create models, it is a one-time activity, and as a result, one can produce a potentially unlimited number of 2D pixel-perfect labeled data samples rendered from the 3D data model. The rendered data ranges from high-quality RGB images to object and class segmentation maps, accurate depth and stereo pairs from multiple camera viewpoints, point cloud data, and many more.

Generating synthetic human models including face and the full human body is even more interesting and relevant, as gathering real human datasets is more challenging than any other kind of data, mainly due to the following limitations:

- The labeling of the human face is especially complex. This includes proper head pose estimation, eye gaze detection, and facial key point detection.
- In most cases, collecting real human data falls under data privacy issues including the General Data Protection Regulation (GDPR).
- Generating 3D scans of the human body with accurate textures requires a complex and expensive full-body scanner and advanced image fusion software.
- The existing real datasets are often biased towards ethnicity, gender, race, age, or other parameters.

This synthetic data can be used for machine learning tasks in several ways:

- Synthetically generated data can be used to train the model directly and subsequently applied the model to real-world data.
- Generative models can apply domain adaptation to the synthetic data to further refine it. A common use case entails using adversarial learning to make synthetic data more realistic.
- Synthetic data can be used to augment existing real-world datasets, which reduces the bias in real data. Typically, the synthetic data will cover portions of the data distributions that are not adequately represented in a real dataset.

In this paper, we propose a pipeline using an open-source tool and a commercially available animation toolkit to generate photo-realistic human models and corresponding ground truths including RGB images and facial depth values. The proposed pipeline can be scaled to produce any number of labeled data samples by controlling the facial animations, body poses, scene illuminations, camera positions, and other scene parameters.

The rest of the paper is organized as follows: Section 2 presents a brief literature review on synthetic virtual human datasets and the motivation against this work. Section 3 explains the proposed framework. Section 4 presents some interesting results and discusses the advantages and future direction of the proposed framework.

TABLE I. REVIEW OF CURRENT SYNTHETIC VIRTUAL HUMAN DATASETS

Dataset	3D Model	Rigged	Full Body	3D Background	Ground Truth
VHuF [1]	Yes	No	No	No	Facial Key points, facial Images, No Depth Data
Kortylewski et al. [3]	Yes	No	No	No	Facial Depth, Facial Images (Only include frontal face with no Complex Background)
Wang et al. [4]	Yes	No	No	No	Facial Image, Head Pose, No depth data
SyRI [5]	Yes	No	Yes	Yes	Full Body Image, No Facial Images
Chen et al. [6]	Yes	No	Yes	No	Body Pose with full body image, No Facial Images
SURREAL [7]	Yes	Yes	Yes	No	Body Pose with Image, Full Body Depth, Optical Flow, No Facial Images
Dsouza et al. [10]	Yes	No	Yes	Yes	Body Pose with Image, Depth including background, Optical Flow, No Facial Images
Ours	Yes	Yes	Yes	Yes	Facial Images, Facial Depth including background, Head Pose

II. RELATED WORK

This section presents an overview of existing 3D virtual human datasets and their applications. It also describes their limitations, which are the main motivation of this work.

Queiroz et al. [1] first introduced a pipeline to generate facial ground truth with synthetic faces using the FaceGen Modeller [2], which uses morphable models to get realistic face skin textures from real human photos. Their work resulted in a dataset called Virtual Human Faces Database (VHuF). VHuF does not contain the ground truth like depth, optical flow, scene illumination details, head pose, and it only contains head models that are not rigged and placed in front of an image as a background. Similarly, Kortylewski et al. [3] proposed a pipeline to create synthetic faces based on the 3D Morphable Model (3DMM) and Basel Face Model (BFM-2017). They only captured the head pose and facial depth by placing the head mesh in the 2D background. The models are not rigged as well. Wang et al. [4] introduced a rendering pipeline to synthesize head images and their corresponding head poses using FaceGen to create the head models and Unity 3D to render images, but they only captured head pose as the ground truth and there is no background. Bak et al. [5] presented the dataset Synthetic Data for person Re-Identification (SyRI), which uses Adobe Fuse CC for 3D scans of real humans and the Unreal Engine 4 for real-time rendering. They used the rendering engine to create different realistic illumination conditions including indoor and outdoor scenes and introduce a novel domain adaptation method that uses synthetic data.

Another common use case of virtual human models is in human action recognition and pose estimation. Chen et al. [6] generated large-scale synthetic images from 3D models and transferred the clothing textures from real images, to predict pose with Convolution Neural Networks (CNN). It only captured the Body Pose as the ground truth. Varol et al. [7] introduced the SURREAL (Synthetic hUMans foR REAL tasks) dataset with 6 million frames with ground truth pose, the depth map, and a segmentation map that showed promising results on accurate human depth estimation and human part segmentation in real RGB images. They used the SMPL [8] (Skinned Multi-Person Linear) body model trained on the CAESAR dataset [9], one of the largest commercially available data that has 3D scans of over 4500 American and European subjects, to learn the body shape and textures, CMU

MoCap to learn the body pose, and Blender to render and accumulate ground truth with different lighting conditions and camera models. Though this is the closest work to this paper that can be found, the human models are not placed in the 3D background, instead, they are rendered using a background image. It also did not capture the Facial Ground Truths as it focused on the full-body pose and optical flow. Dsouza et al. [10] introduced a synthetic video dataset of virtual humans PHAV (Procedural Human Action Videos) that also uses a game engine to obtain the ground truth like RGB images, semantic and instance segmentation, the depth map, and optical flow, but it also does not capture Human Facial Ground truths.

Though there are previous works on creating synthetic indoor-outdoor scenes and other 3D objects, there is limited work done on exploring the existing available open-source tools and other commercially available software to build a large dataset of synthetic human models. Also, another major concern is the realism of the data and per-pixel ground truth. The proposed method tries to fill that gap. It can generate realistic human face data with 3D background and capturing the ground truths like head pose, depth, optical flow, and other segmentation data. As these are fully rigged full-body models, body pose with the other ground truths can also be captured. A detailed featurewise comparison can be found in table 1.

III. METHODOLOGY

This section presents a detailed framework for generating the synthetic dataset including RGB images and the corresponding ground truth.

A. 3D Virtual Humans and Facial Animations

The iClone 7 [11] and the Character Creator [12] software is used to create virtual human models. The major advantages of using iClone and Character Creator are:

- Character Creator provides “Realistic Human 100” models that reduce the bias over ethnicity, race, gender, and age. These pre-built templates can be applied to the base body template as shown in Fig. 1.
- The morphing of different parts of the body can be adjusted to create more variations to the model. Fig. 2 shows adjustment in cheek, forehead, skull, and chin bone.

- Different expressions including neutral, sad, angry, happy, and scared can be added to the models to create facial variations. Fig. 3 presents a sample render of these five expressions from iClone.
- The models provide Physically Based Rendering (PBR) textures (Diffuse, Opacity, Metallic, Roughness) to render high-quality images.
- Models can be exported in different formats (like obj, fbx, and alembic) which are supported by the most popular rendering engines.

Though iClone can render high-quality images, it does not provide the functionality to capture other ground truth data like exact camera locations, head pose, scene illumination details. Therefore, the models were exported from iClone and placed in a 3D scene in the popular free and open-source 3D CG software toolset Blender [13]



Fig. 1. Applying head template on a base female template in Character Creator



Fig. 2. Adjust cheek, forehead, skull and chin bones in Character Creator

B. Model Exporting from iClone

The model created in iClone can be exported in different formats that are supported by the most popular 3D modeling software including Blender. Two of these formats are explored in this work including Alembic (.abc) and FBX (.fbx).

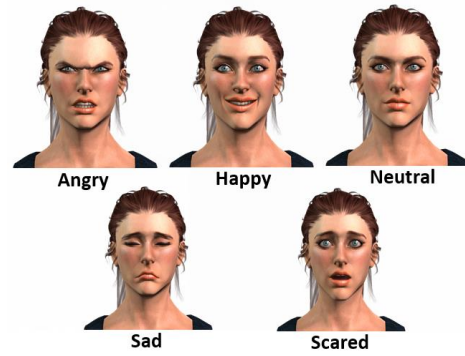


Fig. 3. Sample images with different expression rendered from iClone

In this research, the FBX format is used as it exports the model with proper rigging, which helps to add movements to different body parts including the head. A sample of a fully rigged model is shown in Fig. 4 after the model is loaded in Blender.

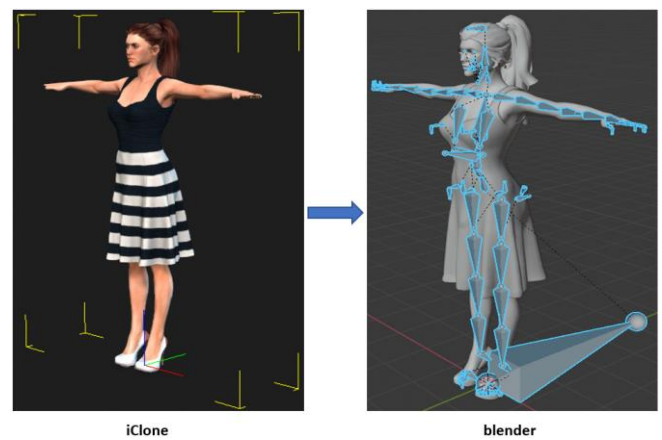


Fig. 4. Sample of a fully rigged model imported in Blender from iClone

C. Rendering

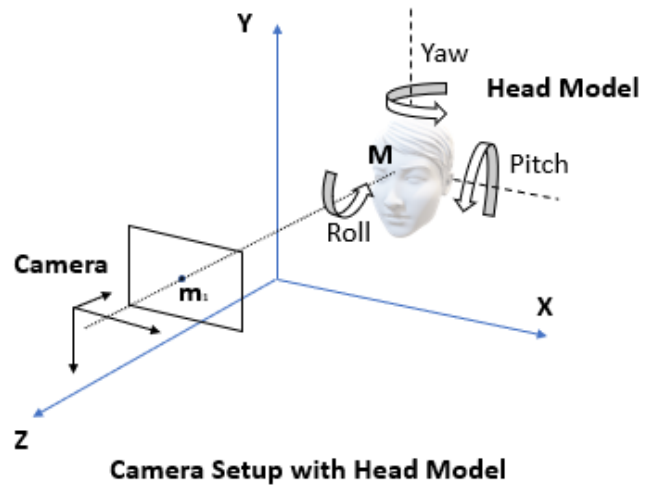
The iClone models are imported to Blender 3D modeling software.

The major components of Blender are Models, Textures, Lighting, Animations, Camera Control (including lens selection, image size, focal length, the field of view (FOV), movement, and tracking), and the rendering engine. The two most common and popular render engines supported by Blender are Cycles and Eevee. Cycles uses a method called path tracing, which follows the path of light and considers reflection, refraction, and absorption to get the realistic rendering, while Eevee uses a method called rasterization, which works with the pixel information instead of paths of light, which makes it fast but reduces the accuracy. A good comparison of these two rendering engines can be found in [14]. A sample workflow of the major components of Blender is described in Fig. 5.

In the current work the following steps are taken to obtain the final output:

- To replicate the process of capturing real data, the camera is placed at a fixed location in the scene and the relative distance from the model to the camera center is varied within a range of 700 mm to 1000 mm to the human model as shown in Fig. 6.

- Different illumination is added to the 3D scene which can be varied to create different realistic lighting which includes point, sun, spotlight, and area light.
- Different render passes are set up in Blender to get the RGB and the corresponding depth images. Cycles rendering engine is used to get a realistic rendering. It has been observed during the rendering of the transparent materials that Cycles path tracing can cause noisy output. To reduce the noise, the branched path tracing is used. It splits the path of the ray as the ray hits the surface and takes into account the light from multiple directions and provide more control for different shaders.
- As the model is rigged, the movement of most of the body parts can be controlled by selecting their bone structure. Here the shoulder and head bones are selected, and the head mesh is rotated with respect to those bones.



Camera Setup with Head Model

Fig. 6. Sample setup of camera and the model

Rotations of yaw (+30 degree to -30 degree), roll (+15 degree to -15 degree), and pitch (+15 degree to -15 degree) are applied to the head and the keyframes are saved. Later these keyframes are used to capture the head pose. A sample setup in Blender is illustrated in Fig. 7.

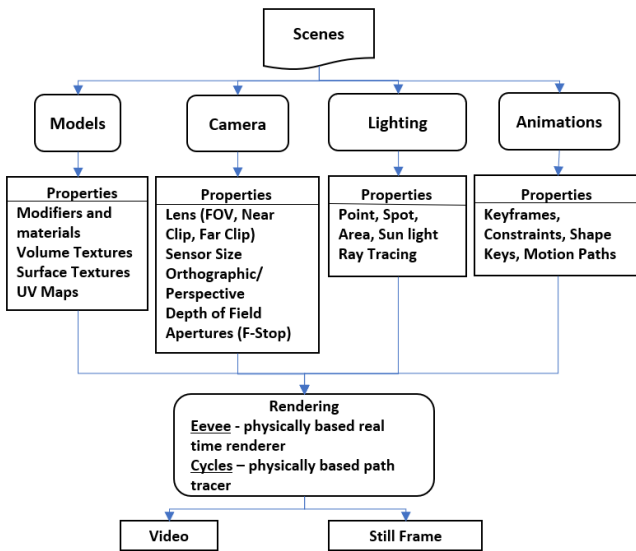


Fig. 5. Sample workflow in Blender

Following the above three steps, the proposed framework works as follows: Using the Real 100 head models a set of virtual human models is created in Character Creator. The texture and morphology of the models are modified to introduce more variations. These models are then sent to iClone where five facial expressions are imposed. The final iClone models with the facial expressions are exported in FBX which consists of the mesh, textures, and animation keyframes.

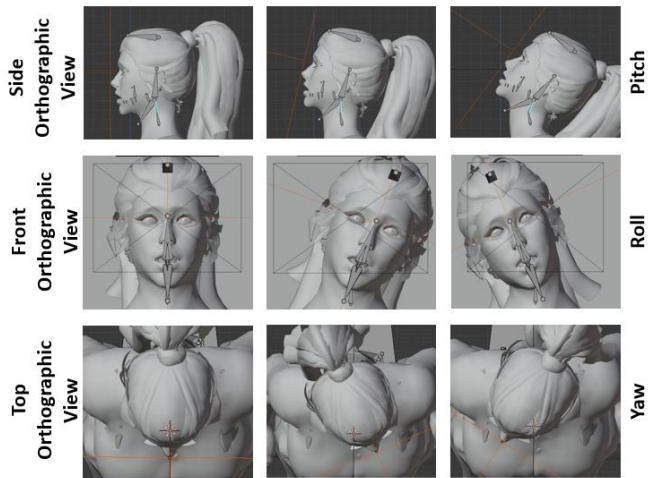


Fig. 7. Applying head movement (yaw, roll, and pitch) on the model in Blender to capture the head pose

The FBX files are then imported and scaled in the Blender world coordinate system. Lights and cameras are added to the scene, whose properties are then adjusted to replicate the real environment. The near and far clip of the camera is set to 0.01 meters and 5 meters respectively. The FOV and the camera sensor size are set to 60 degrees and 36 millimeters respectively. The RGB and Z-pass output of the render layer is then set up in the compositor to get the final result. To apply the rotation, the head and shoulder bone is identified in pose mode and the head mesh is rotated with respect to those bones, and the keyframes are saved. Finally the all the keyframes are rendered to get the RGB and the depth images and the respective head pose (yaw, pitch, and roll) is captured through the python plugin provided by Blender. The overall pipeline is described in Fig. 8.

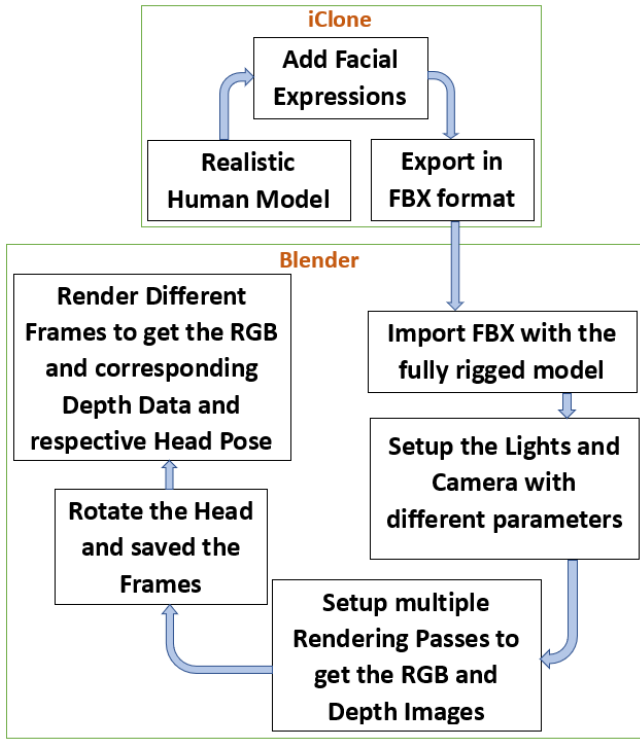


Fig. 8. Pipeline to produce a virtual human

IV. RESULTS AND DISCUSSIONS

Using the framework proposed in Section III, several virtual human models with their corresponding RGB and depth images have been rendered.

The experiments and data generation is performed on an Intel Core i5-7400 3 GHz CPU with 32 GB of RAM equipped with an NVIDIA GeForce GTX TITAN X Graphical Processing Unit (GPU) having 12 GB of dedicated graphics memory. The RGB and depth images are rendered with a resolution of 640 X 480 pixels and their raw depth is saved in .exr format. The average rendering time for each frame is 57.6 seconds. The models are rendered in Blender using different parameters such as the positions of lights, camera parameters, keyframe values of the saved animations. The raw binary depth information and the head pose information are also captured as part of this dataset. Fig. 9 presents the RGB images and their corresponding ground truth depth images (scaled to visualize) with a different head pose. Fig. 10 shows the results with different illuminations. The models then imported to more complex 3D scenes and the ground truth data has been captured. Fig. 11 shows some samples and the corresponding depth with complex backgrounds.

The proposed method allows the creation of potentially unlimited data samples with pixel-perfect ground truth data from the 3D models. Also, the 3D models can be placed in any 3D scene and the data can be rendered within a different environment. Another advantage of using this pipeline of tools is that the positions of the camera and their intrinsic parameters and the scene lighting can be controlled to replicate a real environment. As these models have PBR shading and blender cycle rendering engine utilizes the path ray tracing and accurate bounce lighting the rendered images are more realistic than the previous datasets present. Table 2 provides some samples from other datasets that capture facial synthetic data and shows the result from the proposed model is more realistic and robust than the previous ones. Although

the proposed pipeline can generate a large amount of data more work has to be done in domain transfer and domain adaptation areas to make the images as realistic as possible.



Fig. 9. Sample images of virtual human faces and their ground truth depth (scaled to visualize) with different head pose



Fig. 10. Sample images of virtual human faces in different lighting condition

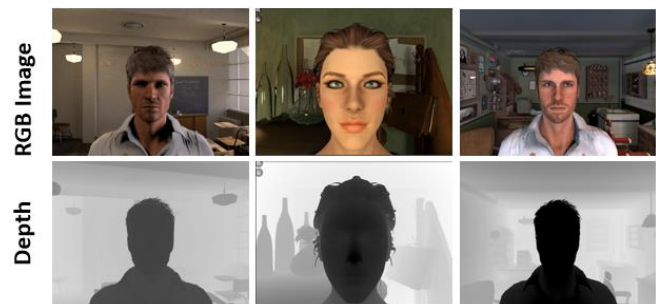






Fig. 11. Sample images and their depth image (scaled to visualize) with more complex background

TABLE II. IMAGE SAMPLES FROM EXISTING FACIAL SYNTHETIC DATASET

Dataset	Ground Truth
VHuF [1]	
Kortylewski et al. [3]	
Wang et al. [4]	
Ours	

V. CONCLUSION

In this work, a framework to synthetically generate a huge set of facial data with variations in environment and facial expressions using available toolchains is explored. This will help to train DNN models, as it covers more variations in expressions and identity. Previously generated synthetic human datasets [6], [7] mostly lack realism and per-pixel ground truth data. The proposed pipeline will help to overcome such limitations. The data generated through this framework can extensively be used for facial depth estimation problems. There are currently a few datasets available with real-world facial images and their corresponding depth [15],[16],[17],[18]. However, it is practically impossible to get pixel-perfect depth images of the human faces due to the limitation of the available sensors like Kinect. The proposed framework can bridge this gap with more accurate ground truth facial depth data. The models can also be used to build more advanced 3D scenes which will cover more complex computer vision tasks such as driver monitoring system, 3D aided face recognition, elderly care, and monitoring.

ACKNOWLEDGMENT

This material is based upon works supported by the Science Foundation Ireland Centre for Research Training in Digitally Enhanced Reality (D-REAL) under grant 18/CRT/6224.

REFERENCES

- [1] Queiroz, R., Cohen, M., Moreira, J. L., Braun, A., Júnior, J. C. J., & Musse, S. R. (2010, August). Generating facial ground truth with synthetic faces. In 2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images (pp. 25-31). IEEE.
- [2] FaceGen Modeller. (n.d.). Retrieved from <http://www.facegen.com/modeller.htm>.
- [3] Kortylewski, A., Egger, B., Schneider, A., Gerig, T., Morel-Forster, A., & Vetter, T. (2019). Analyzing and Reducing the Damage of Dataset Bias to Face Recognition With Synthetic Data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 0-0).
- [4] Wang, Y., Liang, W., Shen, J., Jia, Y., & Yu, L. F. (2019). A deep Coarse-to-Fine network for head pose estimation from synthetic data. *Pattern Recognition*, 94, 196-206.
- [5] Bak, S., Carr, P., & Lalonde, J. F. (2018). Domain adaptation through synthesis for unsupervised person re-identification. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 189-205).
- [6] W. Chen, H.Wang, Y. Li, H. Su, Z.Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen. Synthesizing training images for boosting human 3D pose estimation. 3DV, 2016.
- [7] Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., & Schmid, C. (2017). Learning from synthetic humans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 109-117).
- [8] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6), 248.
- [9] K. Robinette, H. Daanen, and E. Paquet. The CAESAR project: A 3-D surface anthropometry survey. In 3DIM'99
- [10] C. R. d. Souza, A. Gaidon, Y. Cabon, and A. M. Lopez. Procedural generation of videos to train deep action recognition networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2594{2604, July 2017}
- [11] 3D Animation Software: iClone: Reallusion. (n.d.). Retrieved January 27, 2020, from <https://www.reallusion.com/iclone>.
- [12] Character Creator - Fast Create Realistic and Stylized Characters. (n.d.). Retrieved January 27, 2020, from <https://www.reallusion.com/character-creator/>.
- [13] Foundation, B. (n.d.). Home of the Blender project - Free and Open 3D Creation Software. Retrieved January 27, 2020, from <https://www.blender.org/>.
- [14] Lampel, J. (n.d.). Cycles vs. Eevee - 15 Limitations of Real Time Rendering in Blender 2.8. Retrieved January 27, 2020, from <https://cgcookie.com/articles/blender-cycles-vs-eevee-15-limitations-of-real-time-rendering>.
- [15] Rui Min, Neslihan Kose, Jean-Luc Dugelay, "KinectFaceDB: A Kinect Database for Face Recognition," *Systems, Man, and Cybernetics: Systems*, IEEE Transactions on , vol.44, no.11, pp.1534,1548, Nov. 2014, doi: 10.1109/TSMC.2014.2331215.
- [16] N. Erdogmus and S. Marcel, "Spoofing in 2D face recognition with 3D masks and anti-spoofing with Kinect," 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), Arlington, VA, 2013, pp. 1-6.
- [17] Borghi, G., Venturelli, M., Vezzani, R., & Cucchiara, R. (2017). Poseidon: Face-from-depth for driver pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4661-4670).
- [18] Fanelli, G., Weise, T., Gall, J., & Van Gool, L. (2011, August). Real time head pose estimation from consumer depth cameras. In Joint Pattern Recognition Symposium (pp. 101-110). Springer, Berlin, Heidelberg.

Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Data in Brief

journal homepage: www.elsevier.com/locate/dib

Data Article

C3I-SynFace: A synthetic head pose and facial depth dataset using seed virtual human models.



Shubhajt Basak^{a,*}, Faisal Khan^b, Hossein Javidnia^c, Peter Corcoran^b, Rachel McDonnell^d, Michael Schukat^a

^a School of Computer Science, University of Galway, Ireland

^b School of Electrical and Electronics Engineering, University of Galway, Ireland

^c School of Computing, Dublin City University, Ireland

^d School of Computer Science and Statistics, Trinity College Dublin, Ireland

ARTICLE INFO

Article history:

Received 13 February 2023

Revised 9 March 2023

Accepted 17 March 2023

Available online 23 March 2023

Dataset link: [SyntheticFaceDataset_Male_Part2](#)

(Original data);

[SyntheticFaceDataset_Male_Part3](#) (Original

data); [SyntheticHeadPoseDataset_Female_Part1](#)

(Original data);

[SyntheticHeadPoseDataset_Female_Part2](#)

(Original data);

[SyntheticHeadPoseDataset_Female_Part3](#)

(Original data);

[SyntheticHeadPoseDataset_Male_Part1](#)

(Original data);

[SyntheticHeadPoseDataset_Male_Part2](#)

(Original data);

[SyntheticHeadPoseDataset_Female_Part1](#)

(Original data);

[SyntheticHeadPoseDataset_Female_Part2](#)

(Original data);

[SyntheticFaceDataset_Male_Part1](#) (Original

data)

ABSTRACT

This article presents C3I-SynFace: a large-scale synthetic human face dataset with corresponding ground truth annotations of head pose and face depth generated using the iClone 7 Character Creator “Realistic Human 100” toolkit with variations in ethnicity, gender, race, age, and clothing. The data is generated from 15 female and 15 male synthetic 3D human models extracted from iClone software in FBX format. Five facial expressions - neutral, angry, sad, happy, and scared are added to the face models to add further variations. With the help of these models, an open-source data generation pipeline in Python is proposed to import these models into the 3D computer graphics tool Blender and render the facial images along with the ground truth annotations of head pose and face depth in raw format. The datasets contain more than 100k ground truth samples with their annotations. With the help of virtual human models, the proposed framework can generate extensive synthetic facial datasets (e.g., head pose or face depths datasets) with a high degree of control over facial and environmental variations such as pose, illumination, and background. Such large datasets can be

* Corresponding author.

E-mail address: s.basak1@nuigalway.ie (S. Basak).

Social media: [@shubhaBas](#) (S. Basak), [@pcor](#) (P. Corcoran)

<https://doi.org/10.1016/j.dib.2023.109087>

2352-3409/© 2023 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

Keywords:

Synthetic data
 Computer vision
 Virtual human
 Facial depth
 Head pose
 Ground truth data

used for the improved and targeted training of deep neural networks.

© 2023 The Author(s). Published by Elsevier Inc.

This is an open access article under the CC BY license
 (<http://creativecommons.org/licenses/by/4.0/>)

Specifications Table

Subject	Computer Science
Specific subject area	Computer Vision, Head Pose Estimation, Monocular Depth Estimation
Type of data	Images Annotations
How the data were acquired	The initial 3D virtual human models are collected from the low-cost commercially available software iClone [15] and Character Creator [16]. The data has been produced with a 3D graphics rendering pipeline using the open-source Computer Graphics (CG) software Blender [17]. The source of the 3D virtual models and the generating Python scripts are included in this paper.
Data format	Raw
Description of data collection	Rendering - The face images and the corresponding raw depths have been rendered with the following camera parameters - camera near and far the clip is set to 0.001 and 5.0 meters. Camera sensor size and field of view (FOV) are set to 36 millimeters and 60°, respectively. The yaw, pitch, and roll of a head are constrained to $\pm 30^\circ$. To render the face images and raw depths, the RGB and the Z-Pass compositor nodes of the Blender [17] are used with the cycle rendering engine. Background - For complex backgrounds, two scenes (Classroom and Barbershop) from the Blender [17] website have been used.
Data source location	Laboratory: C3Imaging, University of Galway Institution: School of Computer Science, University of Galway City/Town/Region: Galway Country: Ireland
Data accessibility	Direct URL to data: Synthetic Head Pose Datasets: https://data.mendeley.com/datasets/jd4jm3jpp2 [7] https://data.mendeley.com/datasets/mc9fzhkvwp [8] https://data.mendeley.com/datasets/vfrfb56sh4 [9] https://data.mendeley.com/datasets/pttvxjcmpd [10] Synthetic Face Depth Datasets: https://data.mendeley.com/datasets/z4454fyd8b [4] https://data.mendeley.com/datasets/yzjdjj5w39 [5] https://data.mendeley.com/datasets/tbt46rs4y6 [6] https://data.mendeley.com/datasets/33kjk7mj7y [1] https://data.mendeley.com/datasets/5wpj8nh2cv [2] https://data.mendeley.com/datasets/2c2r7998vs [3] Virtual Human Models: Due to licensing issues, we cannot release the virtual human models. But the models can be purchased from the Reallusion website from the following link. These models need to be extracted in fbx format and put in a folder structure as described in the 'Experimental Design' section. https://www.reallusion.com/contentstore/iClone/pack/Realistic_Human_100/default.html
Related research article	Code: https://github.com/shubhajtbasak/blenderDataGeneration S. Basak, P. Corcoran, F. Khan, R. McDonnell and M. Schukat, Learning 3D Head Pose From Synthetic Data: A Semi-Supervised Approach in IEEE Access, vol. 9, pp. 37557-37573, 2021, https://doi.org/10.1109/ACCESS.2021.3063884 [11]

Value of the Data

- The data can be used to train and evaluate computer vision models for head pose estimation, face depth estimation, and face reconstruction. The different lighting conditions and camera positions make the data set robust and capable of generalizing the learning model. Finally, the large scale of the dataset makes it an ideal candidate for deep learning training. For head pose estimation, only two datasets, 300WLP(real) [13] and Nvidia Synhead (synthetic) [14], are available for training. There is no real large-scale dataset available for face depth estimation tasks.
- As the dataset is generated synthetically, both the head pose distribution and the depth data cover a wide range of angles and a wide variation of background which can be challenging to acquire in a constrained laboratory condition.
- Both real head-pose and depth data are acquired by inertial measurement unit (IMU) sensors or depth sensors, which are both prone to sensor noise. For example, often, real depth data has missing depth values or holes in it. The most common head pose dataset, Biwi [12], has an average error of 1 degree [14]. These errors in ground truth data eventually pass to the trained model and affect the model performance. On the contrary, the synthetic head pose and depth data generated by our pipeline are pixel-accurate and do not have any of these issues.
- As both the acquisition of real head pose and face depth data required human subjects, they fall under different data protection and privacy regulations like GDPR, which makes them difficult to collect and use for research purposes. Synthetic data can does not fall under any of these rules, so they are easy to use and generate without any restrictions.
- Apart from the raw data, we have also provided the source for synthetic human models and open-sourced the data generation scripts. Using this code and the open-source CG software Blender [17], one can generate an unlimited amount of pixel-perfect data by changing the camera parameters, scene illumination, and background scenes.

1. Objective

Recent advancement of deep learning makes it the de-facto choice for facial analysis tasks. But the human face has a complex structure and requires high-quality ground truth data to learn the features. Collecting real ground truth 3D face data either requires expensive 3D scanners or depth sensors, which are prone to noise. Also, as this data acquisition involves human subjects, they often fall under data privacy restrictions and other ethical regulations. So, creating synthetic face datasets can be an alternative that can provide the freedom to generate high-quality, large, and diverse face datasets without any such restrictions. A key element of face analysis is the face alignment information as well as the face depth to get the 3D cues. So, in this work, with the help of low-cost 3D human assets and an open-source CG tool, we have created a large face dataset with their corresponding head pose and raw depth annotations. Further, we used this dataset to train a model for head-pose estimation and face-depth estimation to validate the generated synthetic data.

2. Data Description

The data set contains two parts: The synthetic faces with their ground truth depth and another set of synthetic faces with their ground truth head pose annotations. The following section will describe those two parts in detail:

- Face Depth dataset:
Directory Structure - This part of the dataset contains all the rendered face images, their corresponding raw depth in *.exr format, and the head poses data in *.txt files. The root

folder contains two subfolders, 'male' and 'female', with all the identity folders with labels from '0001' to '0045'. Each identity folder has a subfolder called 'Complex', which signifies the complex background of the rendered images. There are two different background scenes in two subfolders; the first folder contains the Barbershop scene, and the second folder has the Classroom scene. Each of these folders contains the ground truth files with five expressions - angry, happy, neutral, sad, and scared. For each expression, there are three different datasets based on the rendered settings. We have collected the ground truth for three different camera and head movements. The 'CameraTran' folder contains the ground truths when the virtual human models are at the origin of the scene, and a random translation motion is added to the camera. The 'HeadCameraRotTran' folder has all the ground truths where the models are in the scene origin, and a head rotation $\{-30^\circ, +30^\circ\}$ is applied to the head bone of the model, and a simultaneous rotation and translation are added to the camera. The last folder, 'HeadRot,' contains the ground truth, where the camera is placed in front of the face in a fixed location, and a rotation $\{-30^\circ, +30^\circ\}$ is applied to the head of the model. Fig. 1a shows the directory structure, and 1b shows some examples of rendered ground truth face images and their corresponding raw normalized depth visualized in grayscale and plasma color maps.

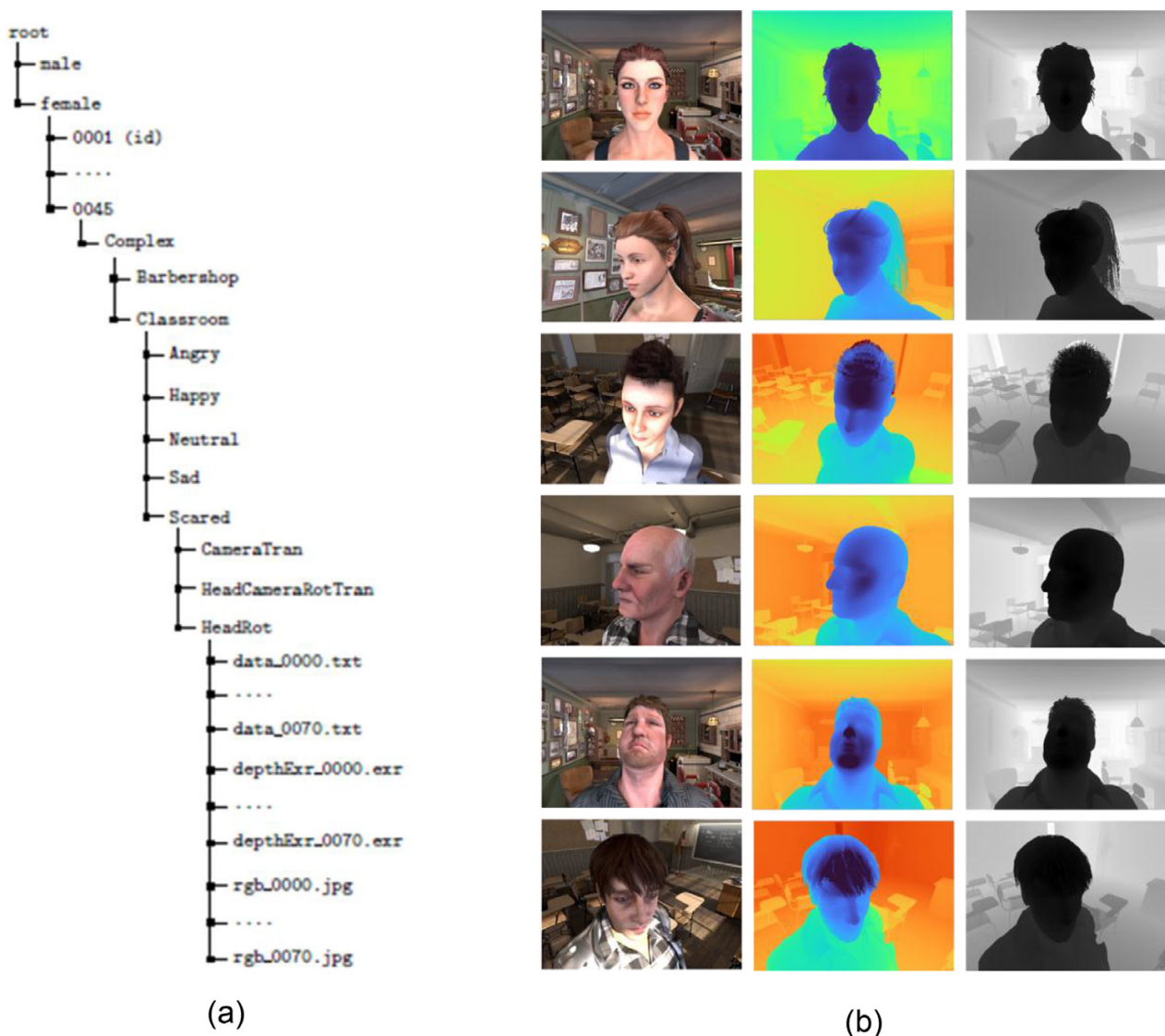


Fig. 1. (a) Folder structure for the facial depth data (b) Sample data rendered in Blender – 1st column is the RGB image, 2nd and 3rd are the normalized depth data (*.exr) visualized in colormap and grayscale.

```

data_0046.txt
1.0 TB Volume /mnt/fastssd/synData/Face...
1 Camera Location: (-0.0002, -0.3136, 1.6779)
2 Head Point Location: (-0.0002, -0.0136, 1.6779)
3 Camera Rotation: Yaw 0.00 Pitch 90.00 Roll -0.00
4 Head Rotation: Yaw -17.90 Pitch 19.43 Roll 2.90
Plain Text Tab Width: 8 Ln 1, Col 1 INS

```

Fig. 2. Sample annotation data consists of the camera position, head point location, camera rotation, and head rotation.

Ground Truth Annotations - Each of these final folders contains three type of ground truth data -

- I. **rgb_<id>.jpg** - The rendered ground truth face image in RGB (*.jpg) format.
- II. **depthExr_<id>.exr** - The raw depth (distance of the face point from the camera center) values in *.exr format.
- III. **data_<id>.txt** - The ground truth annotations in *.txt files. Each text file contains four ground truth annotations - camera location (x,y,z coordinates), head point location (x,y,z coordinates of the head bone), camera rotation (yaw, pitch, and roll of the camera) and head rotation (yaw, pitch, and roll of the head bone). Fig. 2 shows a sample annotation text file.

The depth dataset contains a total of 37670 sets of ground truth images and their corresponding annotations (.exr and .txt) with a total size of around 45 GB.

- Head Pose Dataset:

Directory Structure - This part of the dataset contains the ground truth face images with varying head pose and their corresponding head pose annotations. The root folder contains two subfolders, 'male' and 'female', with all the identity folders with labels from '0001' to '0045'. Each of these folders contain ground truth RGB images and their corresponding head pose data annotations in a text file. Fig. 3a shows the directory structure and 3b shows some samples of ground truth RGB images with varying head poses.

Ground Truth Annotations - Each of these folders contains the ground truth data -

- I. **rgb_<id>.jpg** - The rendered ground truth face image in RGB (*.jpg) format.
- II. **data_<id>.txt** - The ground truth annotations in *.txt files. Similar to the depth data each of these text file contains four ground truth annotations - camera location (x,y,z coordinates), head point location (x,y,z coordinates of the head bone), camera rotation (yaw, pitch, and roll of the camera) and head rotation (yaw, pitch, and roll of the head bone). Fig. 2 shows a sample annotation text file.

This part of the dataset contains a total of 72060 pairs of ground truth images and their corresponding annotations (.txt) with a total size of around 32 GB.

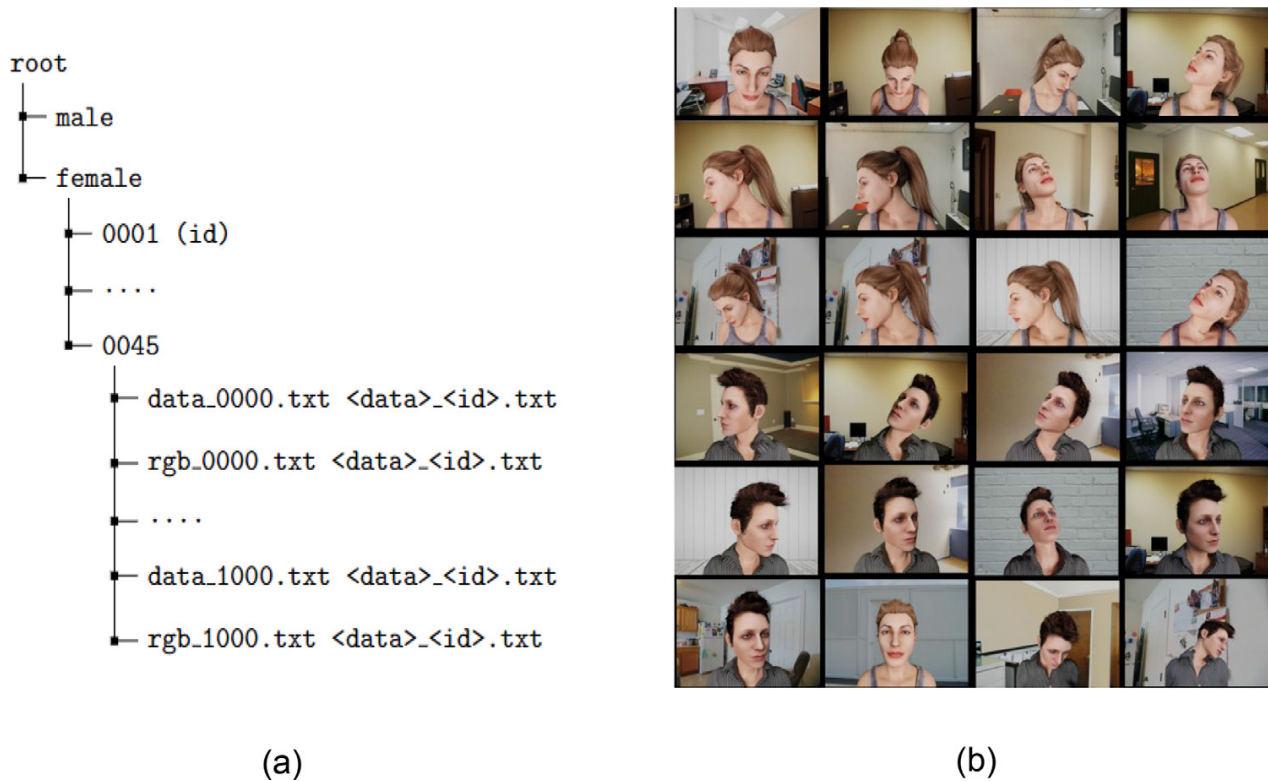


Fig. 3. (a) Folder structure for the head pose data (b) Ground truth synthetic face data rendered in Blender with varying head pose.

3. Experimental Design, Materials and Methods

This section describes the methodology for generating the data set, including the details of the FBX models and the rendering pipeline. It also provides the details of the Python code that has been used to generate the ground truth data with the help of Blender [17].

• 3D Model Generation

Though we are not able to release the virtual model publicly, here we provide the detailed methodology to create this part of the data.

- I. The models can be generated from the ‘Realistic Human 100’ package in iClone [15] software. It provides the functionality to add expressions to the face morphs. The models can be exported in FBX formats from the iClone Character Creator [16].
- II. The iClone [15] tool provides a feature to add different facial expressions to a morph to enhance the facial mesh’s diversity. We have added random changes in the face morph and added different clothing to the model. Then we added four different expressions angry, happy, sad, and scared. The default model is the neutral one.
- III. Then we export the models in fbx format through the iClone Character Creator [16] export pipeline¹.
- IV. The FBX files need to be structured in the following manner to run the python scripts provided to generate the ground truth data. The root folder contains the two subfolders, ‘male’ and ‘female,’ which have the male and female model files within them. In each of these folders, there are identity folders for female and male models, starting from ‘0001’ up to ‘0100’. Each identity folder has a subfolder called ‘Simple’ with five subfolders containing the FBX model files with five different expressions - angry, happy, neutral, sad, and scared. Each of these folders has the

¹ <https://www.reallusion.com/character-creator/blender.html>.

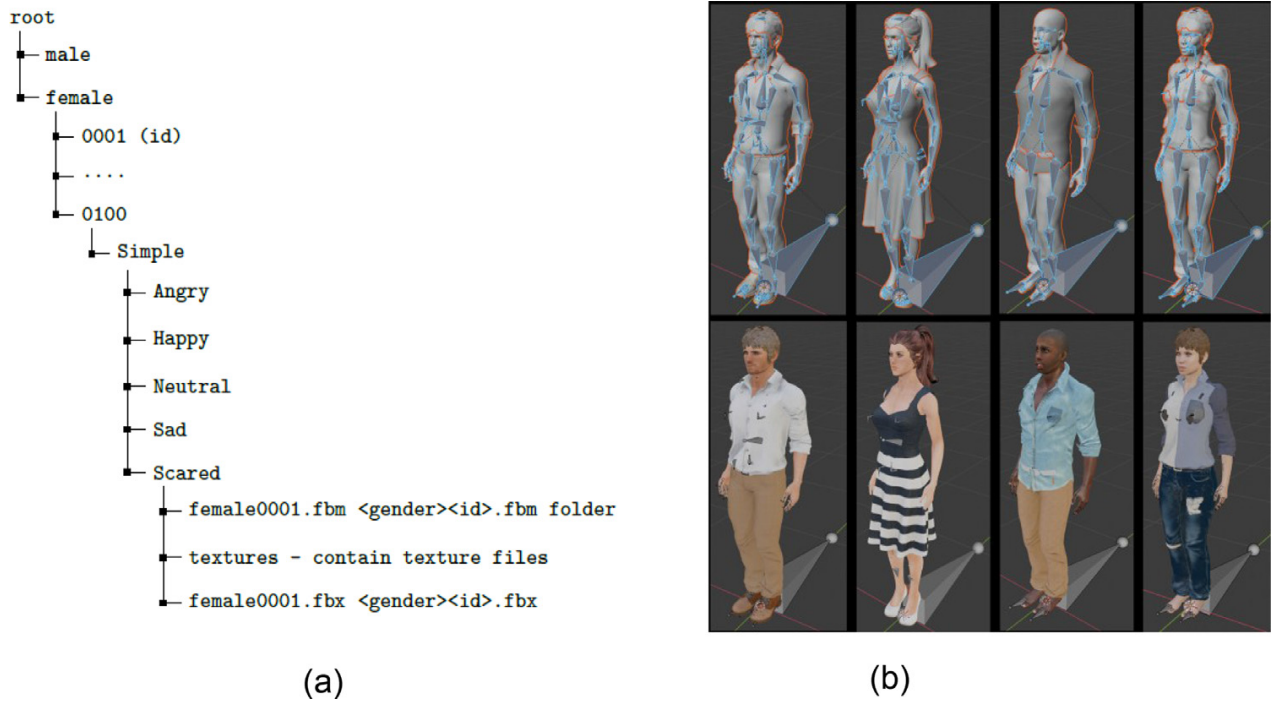


Fig. 4. (a) Folder structure for the virtual human model data (b) Samples of the synthetic human models in fbx format after exporting in Blender.

texture (in textures and fbm folders) and other image files associated with the FBX models. Fig. 4a shows the directory structure of this data set.

V. These models are then imported into the existing Blender scene. We have used two different Blender scenes – Barbershop and Classroom collected from the Blender [17] free library. Fig. 4b shows the sample models (in FBX format) imported into Blender [17] with their armatures (bones) and textures. The imported FBX models are rigged with armatures, later used to add head movements.

• Rendering Ground Truth

To render the ground truth, we followed the following steps. All these steps are performed in batch through python scripts that will be explained in the next section.

- I. A camera is added to the scene in perspective mode, and adequate illumination (e.g., area, sun, point, and spotlight) is added on top of the existing light in the Blender [17] scenes.
- II. To apply the head pose, the neck bone is selected. An empty object is added to the center of the two eyeballs, chosen as the center of the head. The camera's optical axis is set as normal to the plane of the two eyeballs to select the initial head pose. The neck bone's translation and rotation have been copied to the empty object, which adds constraints to the empty object to follow the neck bone.
- III. Once the initial setup is completed, a uniform rotation is applied to the neck bone in the sequence of PRY (pitch, roll, and yaw), and the keyframes are saved.
- IV. After the final design, the ground truth face RGB images with their corresponding raw depths and annotations are generated by a Python script with the help of Blender [17]. Blender's [17] in-built Python support is used to run these scripts. Details about the rendering parameters used while generating the data are shown in Table 1.
- V. We have also provided a separate dataset emphasizing head pose annotations only. We have not used virtual environments because we have not collected the depth information for this part of the data. Instead, we have set real images as the background and put the virtual models in front of them to render the ground truths.

Table 1

Rendering parameters used to generate the ground truth in Blender.

Parameters	Values
Camera center and model head center distance	30 centimeters
Camera Near Clip	0.001 meter
Camera Far Clip	10.0 meters
Camera Sensor Size	36 millimeters
Camera Field of View (FOV)	60 degrees
Blender Rendering Engine	CYCLES
Cycles Progression	BRANCHED PATH
Cycles AA samples	256
Cycles Min transparent bounces	32
Cycles Light sampling threshold	0
Cycles Sample clamp indirect	0
Cycles Max bounces	32
Cycles Diffuse bounces	0
Cycles Glossy bounces	0
Cycles Transparent max bounces	16
Cycles Transmission bounces	16
Rendering Resolution X	640
Rendering Resolution Y	480

- VI. We have added discrete head rotations to the models in an interval of 3° . The yaw, pitch, and roll ranges are $\pm 80^\circ$, $\pm 70^\circ$, and $\pm 55^\circ$, respectively. Though these rotations cover an extensive range of angles, as these are discrete linear sequences, these do not cover some cross-rotation angles. So, to cover all the practical human head pose angles, we have also applied the rotations collected as the ground truth of a real dataset called Biwi [12].
- VII. After applying the head rotation and saving the keyframes, we render each frame through the Blender [17] cycle rendering engine.

- Explanation of Generating Code -

The following section elaborates on the main components of the Python code to generate the ground truth data. The complete code for ground truth generation with complex background is attached to this paper as supplementary material. Also, additional codes for simple and textured background generation can be found on the GitHub page mentioned in the specification table.

importFbx.py: In the first step, the FBX models are imported to the Blender [17] scene (classroom.blend or barbershop.blend), and the imported model is scaled in proper scale to match the Blender [17] scene.

```
bpy.ops.import_scene.fbx(filepath=fbxFilePath)
bpy.context.object.scale = (0.01, 0.01, 0.01)
#save the blend file
bpy.ops.wm.save_mainfile(filepath=blendFilePath)
```

importMisFileBlender.py: First, we set the blender properties. Next, a script is run to add missing texture files (if any) to the *.blend file. sceneSetup.py: In the next step, the Blender [17] scene with the model is set up and head movements are added. Also, the other rendering parameters, like camera properties, are set. In the next step, the midpoint of the two eyeballs is computed by setting empty objects in the eyeball positions and calculating the midpoint of those object locations in global coordinates.

```

emptyList = []
# set the empty at the center of two eye balls
for _obj in _objects:
    if _obj.type == 'MESH':
        if 'Eye' in _obj.name:
            for name in _obj.vertex_groups.keys():
                mt = bpy.data.objects.new("empty", None)
                bpy.context.scene.collection.objects.link(mt)
                mt.name = f"{_obj.name}_{name}"
                emptyList.append(mt.name)
                cl = mt.constraints.new('COPY_LOCATION')
                cl.target = _obj

# get the mid-point of two eyeballs
l_eye = bpy.data.objects[emptyList[1]]
r_eye = bpy.data.objects[emptyList[0]]
l_eye_pos = Point(l_eye.matrix_world.translation)
r_eye_pos = Point(r_eye.matrix_world.translation)
global_location = map_tuple_gen(float,
l_eye_pos.midpoint(r_eye_pos))

```

After setting the midpoint of the eyeballs, which will be the center of the head, another empty object is placed at that point. It will provide the head pose ground truth data. Also, the camera is positioned perpendicular to this point as its initial position.

```

# Create camera object
camera = Camera()
# set the camera in the perspective mode
camera.set_perspective(focal_length=render_focal_length,
sensor=render_sensor_size)

# Offset the camera by the default camera position along the z-axis
camera_location = [global_location[0], -(-global_location[1] +
default_camera_position), global_location[2]]
camera.set_location(camera_location)
camera.set_rotation((radians(90), 0, 0))

# Set the far clipping plane of the camera to a multiple of its
distance from the origin
# far clip around 10 meter
camera.set_far_clipping_plane(default_camera_position * 50)

# create an empty at the center of the two eyeballs
empty1 = bpy.data.objects.new("empty", None)
empty1.location = global_location

```

Finally, the neck bone is chosen on which the head rotation is applied. The rotation is applied while inserting a keyframe to save the animation. The following shows a sample example of adding the animations:

```

pbone = arma.pose.bones['NeckTwist01'] # G6Beta_Neck NeckTwist01
# Set rotation mode to Euler XYZ, easier to understand
# than default quaternions
pbone.rotation_mode = 'XYZ'

# add head rotations and save the frames
for i in range(0, 10):
    pbone.keyframe_insert(data_path="rotation_euler", frame=i + 1)
    pbone.rotation_euler.rotate_axis('Y', radians(-3))

```

compositorSetup.py: Using this script, the compositor nodes in Blender [17] are declared to set the output ground truth paths. It also sets the output data format (e.g., JPEG for RGB and EXR for raw depth data). Finally, the rendering parameters are established, as stated in Table 1.

Following is a code snippet where we set the Blender [17] scene properties.

```

basePath = '../data/Data'
filepath = basePath + '/'.join(bpy.data.filepath.split('/')[:-5:-1])
filepath = filepath.replace('Simple', 'Complex/Barbershop')

# create Render Layer node
renderLayer_node = tree.nodes.new('CompositorNodeRLayers')
renderLayer_node.location = 0, 0

# create Normalize node
normalize_node = tree.nodes.new('CompositorNodeNormalize')
normalize_node.location = 200, 0

# create output node
comp_node = tree.nodes.new('CompositorNodeComposite')
comp_node.location = 400, 0

# link nodes
links = tree.links
links.new(renderLayer_node.outputs[2], normalize_node.inputs[0])
links.new(normalize_node.outputs[0], comp_node.inputs[0])

# create file output node
fileOutput_node = tree.nodes.new('CompositorNodeOutputFile')
fileOutput_node.location = 400, -100
fileOutput_node.base_path = filepath
fileOutput_node.format.color_mode = 'RGB'
fileOutput_node.file_slots[0].path = f'rgb'
fileOutput_node.file_slots[0].format.file_format = 'JPEG'
fileOutput_node.file_slots[0].use_node_format = False

fileOutput_node.file_slots.new("depthExr")
fileOutput_node.file_slots['depthExr'].format.file_format =
'OPEN_EXR'
fileOutput_node.file_slots['depthExr'].format.color_mode = 'RGB'
fileOutput_node.file_slots['depthExr'].format.use_zbuffer = True
fileOutput_node.file_slots['depthExr'].use_node_format = False

# link nodes
links.new(renderLayer_node.outputs[0], fileOutput_node.inputs[0])
links.new(renderLayer_node.outputs[2], fileOutput_node.inputs[1])

```

captureWithGaze.py: Finally, the ground truth is rendered by running this script. As stated in 1.2, the ground truth is generated with three different camera and head rotation settings:

- **Head Rotation:** In this scenario, the camera is placed in its initial location and applied to the frames saved during the above scene setup with the head rotation.

```
#iterate to get the saved key frames
for i in range(0, 70):
    # set the key frame
    bpy.context.scene.frame_set(i)

    # set the generation file path
    bpy.data.scenes["Scene"].render.filepath = dataPathHeadRot +
f'/depth_{i:04d}.png'
    textFilePath = dataPathHeadRot + f'/data_{i:04d}.txt'

    # render
    bpy.ops.render.render(write_still=True)

    # get the camera and the empty (center of head) data
    matrix_empty = bpy.data.objects['empty'].matrix_world
    matrix_camera = bpy.data.objects['Camera'].matrix_world

    # covert to tuple
    r_empty = tuple(map(round,
np.degrees(np.array(matrix_empty.to_euler('XYZ')[0:3])), repeat(4)))
    r_camera = tuple(map(round,
np.degrees(np.array(matrix_camera.to_euler('XYZ')[0:3])),
repeat(4)))

    # write in a text file
    with open(textFilePath, "w") as f:
        f.write('Camera Location: ' + str(tuple(map(round,
matrix_camera.translation, repeat(4)))) + '\n')
        f.write('Head Point Location: ' + str(empty_init_loc) +
'\n')
        f.write('Camera Rotation: ' + 'Yaw %.2f Pitch %.2f Roll
%.2f' % (r_camera[2], r_camera[0], r_camera[1]) + '\n')
        f.write('Head Rotation: ' + 'Yaw %.2f Pitch %.2f Roll %.2f'
% (r_empty[2], r_empty[0], r_empty[1]) + '\n')
```

- **Camera Translation:** In this scenario, the camera is placed in its initial location, and translation is applied while keeping the head model stationary at its initial location. The sample code to apply this translation is given below:

```
for i in range(0, 1):
    # set camera initial location and rotation
    camera.location = camera_init_loc
    camera.rotation_euler = camera_init_rotation

    # apply uniform translation to the camear in the Y plane
    camera.location.x = camera.location.x + random.uniform(0.001,
0.140)
    camera.location.z = camera.location.z + random.uniform(0.001,
0.04)
```


- **Head Rotation and Camera Rotation & Translation:** In this scenario, firstly, a virtual half sphere centering the empty object (center of the eyeballs) is constructed before randomly generating distributed points on that half sphere to where the camera is moved. In contrast, the camera axis is pointed to the empty object. At the same time, the previously saved frames are applied to the head model to rotate the head linearly. The sample code to generate the points in a half sphere and apply them to the camera position is shown below:

```

from sympy.geometry import Point
def point_at(obj, target, roll=0):
    """
        Rotate obj to look at target

        :arg obj: the object to be rotated. Usually the camera
        :arg target: the location (3-tuple or Vector) to be looked
at
        :arg roll: The angle of rotation about the axis from obj to
target in radians.

        Based on: https://blender.stackexchange.com/a/5220/12947
(ideasman42)
    """
    if not isinstance(target, mathutils.Vector):
        target = mathutils.Vector(target)
    loc = obj.location
    # direction points from the object to the target
    direction = target - loc

    quat = direction.to_track_quat('-Z', 'Y')
    quat = quat.to_matrix().to_4x4()
    rollMatrix = mathutils.Matrix.Rotation(roll, 4, 'Z')

    # remember the current location, since assigning to
obj.matrix_world changes it
    loc = loc.to_tuple()
    obj.matrix_world = quat @ rollMatrix
    obj.location = loc

```

```

def generatePoints(r, center, num):
    """
        Generate the cartesian co-ordinates of random points on a
        surface of sphere
        Constraints : phi - with (60,120) degree
                    theta - with (-45,45) degree

        :arg r: radius of the sphere
        :arg center: center of the sphere
        :arg num: number of random locations to be generated

        Based on:
https://stackoverflow.com/questions/33976911/generate-a-random-sample-of-points-distributed-on-the-surface-of-a-unit-sphere
    """

    pts = []
    for i in range(0, num):
        phi = radians(random.uniform(60, 120))
        theta = radians(random.uniform(-45, 45))
        x, y, z = (center.x - sin(theta) * sin(phi) * r), (center.y
- (cos(theta) * sin(phi) * r)), (
                    center.z + cos(phi) * r)
        pts.append((x, y, z))
    return pts

center = Point(tuple(empty_init_loc))
r = 0.3
num = 70
pts = generatePoints(r, center, num)
start = 0
for i, pt in enumerate(pts, start):
    frame = i
    # set heead rotation frame
    bpy.context.scene.frame_set(frame)
    bpy.context.view_layer.update()

    # apply camera location
    camera.location = tuple(map(float, tuple(pt)))
    # point the camera to the empty
    point_at(camera, empty_init_loc, roll=radians(0))

```

renderFromCMD.py: To run correctly, we must execute these individual scripts within the Blender [8] python console. So, to generate the ground truth in batch, we pass these individual scripts as a command line argument to the Blender [17] executable while iterating through all the fbx files from the model root folder. We execute the scripts in the same order as discussed above - importFbx.py → importMisFileBlender.py → sceneSetup.py → compositorSetup.py → captureWithGaze.py. A sample execution (for importFbx.py) is shown in the following code snippet.


```

import os
# set all the directories
target_folder = r'blenderDataGenerationFinal/data/FullBodyModels' #
fbx model path
blenderScenePath =
r'blenderDataGenerationFinal/data/Environments/Barbershop/barbershop_i
nterior_gpu.blend' # blender scene path
sceneName = '_barbershop'
blenderPath = '/blender_path/blender-2.81-linux-glibc217-
x86_64/blender ' # blender executable path
importfbxScript = r'\importFbx.py'
# iterate through all the fbx files
for root, dirs, files in os.walk(target_folder):
    for dir in dirs:
        for file in os.listdir(str(root) + '/' + str(dir)):
            if file.endswith(".fbx"):
                fbxFFilePath = os.path.join(root, dir, file)
                blenderFile =
os.path.basename(fbxFFilePath).split('.')[0] + sceneName + '.blend'
                blenderFilePath = os.path.join(root, dir, blenderFile)
                if os.path.exists(blenderFilePath):
                    os.remove(blenderFilePath)
                pwd = 'XXXXXXXX'
                cmd = blenderPath + blenderScenePath + " --background
-P " \
                    + importfbxScript + " " + fbxFFilePath + " " +
blenderFilePath
                # call blender executable with the required arguments
                p = subprocess.call('echo {} | sudo -S {}'.format(pwd,
cmd), shell=True)

```

Ethics Statements

The work did not involve human or animal subjects or data from social media platforms.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

[SyntheticFaceDataset_Male_Part2 \(Original data\)](#) (Mendeley Data).
[SyntheticFaceDataset_Male_Part3 \(Original data\)](#) (Mendeley Data).
[SyntheticHeadPoseDataset_Female_Part1 \(Original data\)](#) (Mendeley Data).
[SyntheticHeadPoseDataset_Female_Part2 \(Original data\)](#) (Mendeley Data).
[SyntheticHeadPoseDataset_Female_Part3 \(Original data\)](#) (Mendeley Data).
[SyntheticHeadPoseDataset_Male_Part1 \(Original data\)](#) (Mendeley Data).
[SyntheticHeadPoseDataset_Male_Part2 \(Original data\)](#) (Mendeley Data).
[SyntheticHeadPoseDataset_Female_Part1 \(Original data\)](#) (Mendeley Data).
[SyntheticHeadPoseDataset_Female_Part2 \(Original data\)](#) (Mendeley Data).
[SyntheticFaceDataset_Male_Part1 \(Original data\)](#) (Mendeley Data).

CRedit Author Statement

Shubhajt Basak: Conceptualization, Methodology, Software, Writing – original draft; **Faisal Khan:** Data curation, Resources, Investigation; **Hossein Javidnia:** Conceptualization, Formal analysis, Investigation; **Peter Corcoran:** Formal analysis, Validation, Supervision; **Rachel McDonnell:** Supervision, Funding acquisition; **Michael Schukat:** Writing – review & editing, Supervision, Project administration.

Acknowledgments

Funding: This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (D-REAL) under Grant No. 18/CRT/6224.

Supplementary Materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.dib.2023.109087](https://doi.org/10.1016/j.dib.2023.109087).

References

- [1] S. Basak, H. Javidnia, F. Khan, R. Mc Donnell, P. Corcoran, M. Schukat, "Syntheticfacedataset_male_Part1", Mendeley Data V1 (2022), doi:[10.17632/33kjk7mj7y.1](https://doi.org/10.17632/33kjk7mj7y.1).
- [2] S. Basak, H. Javidnia, F. Khan, R. Mc Donnell, P. Corcoran, M. Schukat, "Syntheticfacedataset_male_Part2", Mendeley Data V1 (2022), doi:[10.17632/5wpj8nh2cv.1](https://doi.org/10.17632/5wpj8nh2cv.1).
- [3] S. Basak, H. Javidnia, F. Khan, R. Mc Donnell, P. Corcoran, M. Schukat, "Syntheticfacedataset_male_Part3", Mendeley Data V1 (2022), doi:[10.17632/2c2r7998vs.1](https://doi.org/10.17632/2c2r7998vs.1).
- [4] S. Basak, H. Javidnia, F. Khan, R. Mc Donnell, P. Corcoran, M. Schukat, "Syntheticfacedataset_female_Part1", Mendeley Data V1 (2022), doi:[10.17632/z4454fyd8b.1](https://doi.org/10.17632/z4454fyd8b.1).
- [5] S. Basak, H. Javidnia, F. Khan, R. Mc Donnell, P. Corcoran, M. Schukat, "Syntheticfacedataset_female_Part2", Mendeley Data V1 (2022), doi:[10.17632/yzjdjj5w39.1](https://doi.org/10.17632/yzjdjj5w39.1).
- [6] S. Basak, H. Javidnia, F. Khan, R. Mc Donnell, P. Corcoran, M. Schukat, "Syntheticfacedataset_female_Part3", Mendeley Data V1 (2022), doi:[10.17632/tbt46rs4y6.1](https://doi.org/10.17632/tbt46rs4y6.1).
- [7] S. Basak, H. Javidnia, F. Khan, R. Mc Donnell, P. Corcoran, M. Schukat, "Syntheticheadposedataset_male_Part1", Mendeley Data V2 (2022), doi:[10.17632/jd4jm3jpp2.2](https://doi.org/10.17632/jd4jm3jpp2.2).
- [8] S. Basak, H. Javidnia, F. Khan, R. Mc Donnell, P. Corcoran, M. Schukat, "Syntheticheadposedataset_male_Part2", Mendeley Data V2 (2022), doi:[10.17632/mc9fzhkvwp.2](https://doi.org/10.17632/mc9fzhkvwp.2).
- [9] S. Basak, H. Javidnia, F. Khan, R. Mc Donnell, P. Corcoran, M. Schukat, "Syntheticheadposedataset_female_Part1", Mendeley Data V2 (2022), doi:[10.17632/vfrfb56sh4.2](https://doi.org/10.17632/vfrfb56sh4.2).
- [10] S. Basak, H. Javidnia, F. Khan, R. Mc Donnell, P. Corcoran, M. Schukat, "Syntheticheadposedataset_female_Part2", Mendeley Data V2 (2022), doi:[10.17632/pttvxjcmpd.2](https://doi.org/10.17632/pttvxjcmpd.2).
- [11] S. Basak, P. Corcoran, F. Khan, R. McDonnell, M. Schukat, Learning 3d head pose from synthetic data: a semi-supervised approach, *IEEE Access* 9 (2021) 37557–37573, doi:[10.1109/ACCESS.2021.3063884](https://doi.org/10.1109/ACCESS.2021.3063884).
- [12] G. Fanelli, M. Dantone, J. Gall, A. Fossati, L. Van Gool, Random forests for real time 3d face analysis, *Int. J. Comput. Vis.* 101 (3) (2013) 437–458.
- [13] X. Zhu, Z. Lei, X. Liu, H. Shi, S.Z. Li, Face alignment across large poses: a 3d solution, in: *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, 2016, pp. 146–155.
- [14] J. Gu, X. Yang, S. De Mello, J. Kautz, Dynamic facial analysis: from bayesian filtering to recurrent neural network, in: *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, 2017, pp. 1548–1557.
- [15] Real-Time 3D Animation Software | iClone | Reallusion. Accessed: Feb. 24, 2023. [Online]. Available: <https://www.reallusion.com/iclone/>.
- [16] Character Creator—Fast Create Realistic and Stylized Characters. Accessed: Feb. 24, 2023. [Online]. Available: <https://www.reallusion.com/character-creator/>.
- [17] Blender—Home of the Blender Project—Free and Open 3D Creation Software. Accessed: Feb. 24, 2023. [Online]. Available: <https://www.blender.org/>.

Chapter 3

Head Pose and Facial Depth Estimation using synthetic Facial Data

3.1 Background

Recent computer graphics technology advancements have made synthetic data a popular alternative to real data in any deep learning-based computer vision task. Specifically, when it comes to facial analysis, collecting real-world data often suffers from privacy and ethical concerns. Even though with the availability of virtual human models, we can create synthetic human datasets, the realism of the synthetic data remains an issue when it comes to the performance of the trained model. As the synthetic rendered face images do not look exactly like the real human face, the domain gap between the real and synthetic faces reduces the model accuracy in any facial analysis task. To reduce the domain gap between the synthetic and real domains, the most popular approach is to apply for knowledge transfer from the synthetic domain to the real domain. Nowruzi et al. [104] published a detailed study on the application of synthetic data in object detection tasks and drew conclusions regarding the best use of synthetic data in object detection tasks. They tested two different approaches for transfer learning using hybrid datasets (mixing of real and synthetic data):

- Synthetic-real data mixing - where a small amount of real data is mixed with a large synthetic dataset, and the model is trained on the hybrid dataset.
- Fine-tuning on real data - where we first train the model with only synthetic data and then fine-tune the previously trained model on a small portion of real data.

The main finding of [104] is that fine-tuning on real data performs significantly better than mixing real and synthetic data.

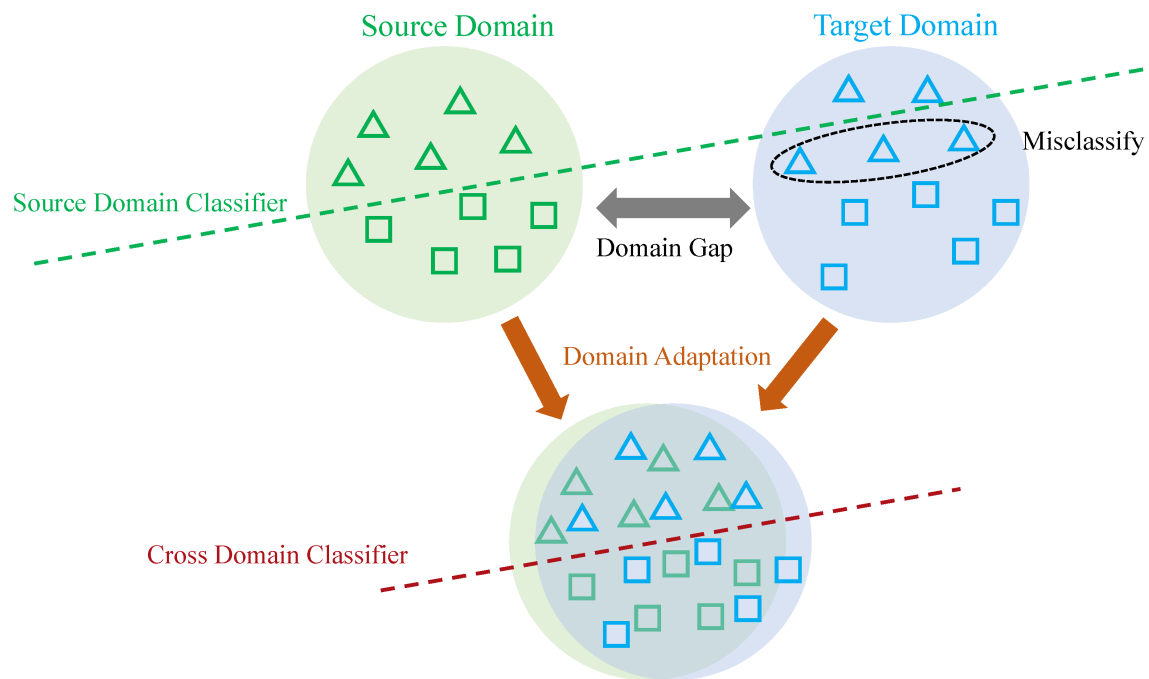


Fig. 3.1 Applying domain adaptation to train a cross-domain classifier [123]

Though these data fusion-based transfer learning approaches make synthetic data more popular in deep learning tasks, there is another set of approaches under transfer learning known as domain adaptation, where the model is trained in one domain of data to work well on a different target domain. Here the source and target domain both have the same feature space but of different distribution in contrast to other transfer learning approaches where the feature space of the target domain differs from the feature space of the source domain. So domain adaptation methods are a natural fit for synthetic data, where we would like to train a model in the source domain of synthetic data and expect the model to work well in the target domain of real data. Specifically, feature-level or model-level domain adaptation is more relevant while working with synthetic data. Here the method works in feature space or model weights to train the network so that it simultaneously learns the common features from both the real and synthetic domains while learning the actual objective. A demonstration of the domain adaptation is presented in figure 3.1, where it is shown how the domain gap is being reduced to align the label spaces of the source and target domain for a classification problem. A major contribution towards model-level domain adaptation was

made by Ganin, and Lempitsky [52], who introduced a generic framework for unsupervised domain adaptation.

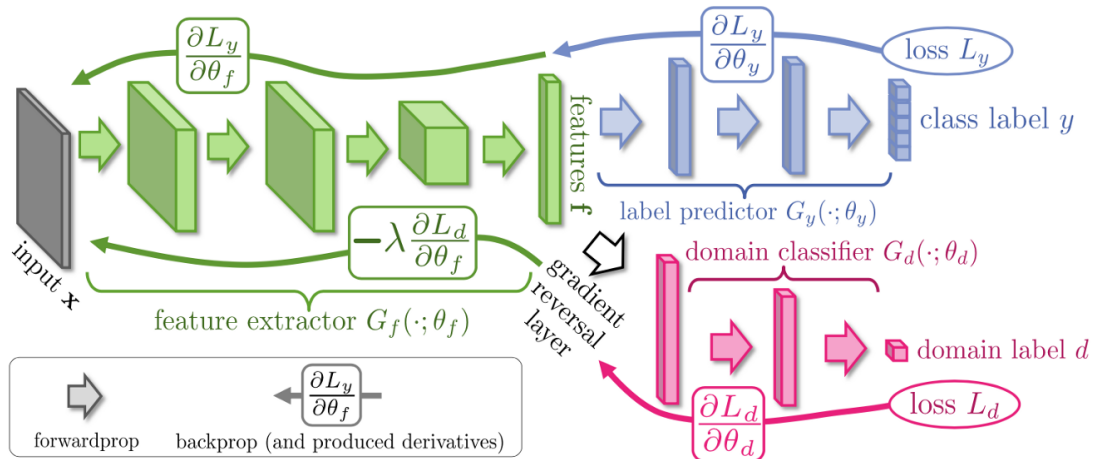


Fig. 3.2 Generic training framework for unsupervised domain adaptation introduced by Ganin and Lempitsky [52]

The approach proposed by [52] consists of three components as shown in figure 3.2:

- The feature extractor (G_f) - is responsible for extracting the features from the visual input.
- The label predictor (G_y) - performs the necessary task (e.g., classification). This will be used during the inference.
- The domain classifier (G_d) - takes the same features extracted by the feature extractor from the source and a target domain and tries to classify them.

They proposed a joint training strategy to train the label predictor for the main task and simultaneously try to make the domain classifier perform as badly as possible. This is achieved by adding a gradient reversal layer as shown in figure 3.2 in the back-propagation path of the domain classifier by multiplying the gradient with a negative constant value which acts as a weight. The value of this constant is selected by empirical study. This way, the feature extractor learns in such a way that the labels of the source and target domain mix up properly, as shown in figure 3.1, and improves the domain adaptation performance. In our task here, the source domain is the synthetic data, and the target domain is the real data set.

3.2 Research Objective

As stated in chapter 2, with the help of the CG toolchain, we have generated and published a large synthetic face dataset and their corresponding ground truth, which consists of head pose and facial depth annotations. So the main objective of this study is to validate the effectiveness of the data generated by our proposed pipeline for deep learning tasks. We have considered two computer vision tasks: head pose estimation (HPE) from a single headshot image and monocular facial depth estimation. Though there are previous works on visual tasks like object detection and semantic segmentation [104, 79, 135], which use synthetic training data for data augmentation and other kinds of transfer learning methods, there are only a very limited amount of studies available for HPE which utilize synthetic training data. Also, as per our best knowledge, no previous work has studied depth estimation tasks of human faces that use synthetic data. So to validate the synthetic data, we conducted the following:

- We investigated the performance of the current SOTA HPE model when solely trained on our synthetic head pose data and compared the performance with the other available synthetic datasets.
- We applied the data fusion and fine-tuning-based transfer learning strategy of training the model and investigating the model performance.
- We also examined the potential of the unsupervised synthetic-to-real domain adaptation methods in HPE tasks with the help of our synthetic dataset and a small subset of the Biwi datasets.
- Finally, we explored the potential of our synthetic face data and the captured raw depth data in monocular facial depth estimation tasks.

3.3 Summary of Contribution

In order to accomplish the above-discussed objectives, we worked on the two main tasks of HPE and facial monocular depth estimation. The following subsections will discuss the contributions with respect to these two tasks.

3.3.1 Learning Head Pose from synthetic Data through Regression

This work is presented in the conference paper - Basak, Shubhajit, Faisal Khan, Rachel McDonnell, and Michael Schukat. "Learning accurate head pose for consumer technology

from 3D synthetic data." In 2021 IEEE International Conference on Consumer Electronics (ICCE), pp. 1-6. IEEE, 2021. A copy of the paper is attached at the end of this chapter.

The contributions of the authors for the above-mentioned research work [16] as per the four major criteria discussed in section 1.4 is presented in the table 3.1.

Table 3.1 Author's Contribution to [16]

Contribution Criteria	Contribution Percentage
Ideation	SB 90%,FK 10%
Experiments & Implementations	SB 90%,FK 10%
Manuscript Preparation	SB 90%,RM 5%, MS 5%
Background Work	SB 70%,MS 20%,RM 10%

As stated in the previous section, to validate the usage of the synthetic head pose generated by our method, we trained the SOTA HPE model FSA-Net [158]. This model is based on feature aggregation and soft stagewise regression, which employs a coarse-to-fine strategy for classification followed by regression. We trained the model solely on our synthetic data and tested it on the Biwi dataset. We use a popular face recognizer (FR) MTCNN to exclude some of the frames of the Biwi dataset, which have extreme angles, where the FR is not able to detect the face in it, to create the test dataset. We have also conducted a detailed ablation study varying the training dataset's yaw, pitch, and roll angles and presented the results in the subsequent work [12]. The results show that training the network solely on our synthetic data is able to achieve neat SOTA performance when tested on Biwi dataset. We further filtered the Biwi on a narrower angle of Yaw(+60, -60), Pitch(+60, -60), and Roll(+10, -10); the results are even better than the nearest comparable work, which used a mix of real and synthetic data as their training set.

3.3.2 Learning Head Pose from synthetic Data through adversarial Domain Adaptation

The previous work is further extended by applying the unsupervised domain adaptation and is presented through the article - Basak, Shubhajit, Peter Corcoran, Faisal Khan, Rachel McDonnell, and Michael Schukat. "Learning 3D head pose from synthetic data: A semi-supervised approach." IEEE Access 9 (2021): 37557-37573. A copy of the paper is attached at the end of this chapter.

The contributions of the authors for the above-mentioned research work [12] as per the four major criteria discussed in section 1.4 is presented in the table 3.2.

Table 3.2 Author's Contribution to [12]

Contribution Criteria	Contribution Percentage
Ideation	SB 100%
Experiments & Implementations	SB 90%,FK 10%
Manuscript Preparation	SB 80%,PC 10%,RM 5%, MS 5%
Background Work	SB 70%,MS 20%,PC 10%

The previous initial work on learning head pose solely from synthetic data is further extended in this work. As found in the previous work, a SOTA model trained solely on our synthetic data is able to outperform the nearest method that uses synthetic data in a narrower range of head poses. So in this work, our goal is to improve the model performance further while using synthetic data only. We proposed to introduce unsupervised domain adaptation via adversarial learning to the HPE task. Almost all of the previous studies that follow [52] and apply for domain adaptation work on classification tasks where they consider partially shared label spaces. These assume identical or shared label spaces where for every sample of the target data, there exists a source data with the same label class. However, in the real world, this assumption does not fit as there exist only a very small amount of real data (target domain) compared to synthetic data (source domain). So while training the domain adaptation as per [52], the source and target labels are tried to align with each other, but as the target label space is not matched with source labels, it causes a negative transfer. But since HPE label spaces are continuous distributions, this proposed method cannot be applied directly to the HPE problem. So we first introduced an adversarial learning module to a SOTA regressor based on [52] and then proposed a specific training methodology, which was able to sample out the nearest data from the target dataset to pass through the adversarial training to reduce the negative transfer effect. More detailed description can be found in section VII-B of the paper [12].

3.3.3 Monocular Facial Depth Estimation from synthetic Images

The initial work is presented in the conference paper - Khan, Faisal, Shubhajit Basak, and Peter Corcoran. "Accurate 2D facial depth models derived from a 3D synthetic dataset." In 2021 IEEE International Conference on Consumer Electronics (ICCE), pp. 1-6. IEEE, 2021. Subsequent detailed work is then presented in the article - Khan, Faisal, Shahid Hussain, Shubhajit Basak, Joseph Lemley, and Peter Corcoran. "An efficient encoder–decoder model for portrait depth estimation from single images trained on pixel-accurate synthetic data." *Neural Networks* 142 (2021): 479-491. A copy of these papers are attached at the end of this chapter 3.5.

The contributions of the authors for the above-mentioned research work [84, 86] as per the four major criteria discussed in section 1.4 is presented in table 3.3. Though the primary work for these was carried out by Faisal Khan, my contribution to these work are:

- Preparing the training data generated by the methodology discussed in chapter 2. Cleaning the data and preparing the data loader for the network.
- Proposing the basic structure of the lightweight U-Net architecture. More details can be found in section 5.1 in [86].
- Proposing and implementing the hybrid loss function utilizing five subfunctions.

Table 3.3 Author’s Contribution to [84, 86]

Contribution Criteria	Contribution Percentage
Ideation	FK 70%,SB 20%,JL 10%
Experiments & Implementations	FK 70%,SB 30%
Manuscript Preparation	FK 70%,SH 20%,SB 10%
Background Work	FK 70%,PC 30%

We used the facial ground truth depth data in the depth estimation task to accomplish the final objective. In the initial work [85, 84], we proposed a shallow U-Net-based encoder-decoder model with conventional loss functions. We divided our synthetic dataset into train and test sets and evaluated our model. We also compared the results of replacing the encoder network with other SOTA feature extractors like Resnet, EfficientNet, etc., and building the decoder with a basic block of CNN layers concatenated by bilinear upsampling layers. In the subsequent work [86], we have extended the previous works and presented a detailed study of our synthetic data for facial depth estimation tasks. In this work, we proposed a hybrid multi-task loss function that consists of point-wise loss, gradient loss, surface normal loss, and structural similarity index measure (SSIM) loss. The influence of each loss term on the overall loss performance is managed by adding weight to each of these terms. The weights of each of these loss terms are set empirically through ablation study. We also used a lightweight auto-encoder model, which incorporates a two-stage mechanism. The encoder consists of a Mobilenet-based depthwise decomposition mechanism. In the decoder layer, the final high-resolution output depth is predicted by five upsampling layers and a single pointwise layer. The proposed model combined with the hybrid loss shows performance equal to, or better than, current SOTA depth estimation networks while being more computationally efficient than others.

3.4 Discussion on Contribution

This work provides a solid understanding of how synthetic data can be used with a limited amount or without real data in different complex high-level computer vision tasks like head pose estimation and monocular depth estimation. Through a detailed review and experiments, it is clear that the main bottleneck in most machine-learning solutions is the availability of clean and accurately annotated data. Collecting accurate head pose and facial depth data with enough variations is almost impossible as well as expensive because of the limitations of sensors and other environmental and ethical constraints. Synthetic data generated from open-source CG tools can be a viable solution.

As discussed in chapter 1, some of the previous works argue that the realism of the synthetic data may not be very important for the performance of the deep learning model. But specific to the facial analysis, we have shown that the realism and the background of the scene play an important role in the model performance and accuracy. The data fusion-based transfer learning method is expected to improve the model performance. But through experiments, we have found that adding a very small amount of real data with our synthetic data gives better results compared to the previous synthetic data-based HPE method. Specifically, the previous method [148] used a set of 12k real data (from Biwi) and 208k of synthetic data, while we have used only 1k of real data (from Biwi) and 300k of our synthetic data during the model training. So while using a comparatively very small amount (only 8%) of real data, we are able to achieve a better result by reducing the mean error from 4.76 to 4.62 in yaw, from 5.48 to 4.537 in pitch and from 4.29 to 3.33 in the roll. We also trained on only synthetic data while replacing the plain background with a mix of real and textured images, which reduced the mean average error from 7.13 to 6.34. Through the proposed adversarial domain adaptation training, the performance of the network is further improved by reducing the mean average error from 6.34 to 5.13. Though the unsupervised domain adaptation from synthetic to the real domain is very popular for classification tasks, the traditional pipeline is difficult to apply in regression which has continuous values to predict. So we proposed an alternative methodology for training where we split the target domain into bins without any direct supervision and applying the adversarial training to the source (synthetic) and target (real) domain keeping them in the same bin.

While experimenting with the synthetic depth data, we have shown that by selecting the proper weighting scheme in a multi-loss function, a comparatively lightweight autoencoder model can achieve an equal or better result than the current SOTA models. As there are no previous methods available that train a monocular facial depth estimation model on synthetic data, we have published a new benchmark for single-frame facial depth estimation from the synthetic face. At 16.41 G-MACs per frame, this approach can enable real-time single-frame

depth estimation. Though through the experimental results, we have found that the depth estimation approach can estimate the average or mean shape of the face quite well, we are not able to predict the detailed shape with high accuracy. This fact encouraged us to move to predict the detailed face reconstruction from a single monocular real human face. We will discuss this in the next chapter, where we have worked on estimating the face shape from a single face image.

Overall, through this work, we have found that synthetic data gives promising results on facial analysis tasks and can be a valuable alternative to real data. Particularly synthetic data remains important for reducing the effect of dataset bias in real datasets, covering corner cases, or taking care of a problem in different modalities like FR in the thermal or infrared domain.

3.5 Copy of Published Works

Received February 4, 2021, accepted February 22, 2021, date of publication March 4, 2021, date of current version March 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063884

Learning 3D Head Pose From Synthetic Data: A Semi-Supervised Approach

SHUBHAJIT BASAK¹, PETER CORCORAN², (Fellow, IEEE), FAISAL KHAN²,
RACHEL MCDONNELL³, AND MICHAEL SCHUKAT¹, (Member, IEEE)

¹School of Computer Science, National University of Ireland Galway, Galway, H91 TK33 Ireland

²Department of Electronic Engineering, College of Science and Engineering, National University of Ireland Galway, Galway, H91 TK33 Ireland

³School of Computer Science and Statistics, Trinity College Dublin, Dublin 2, D02 PN40 Ireland

Corresponding author: Shubhajit Basak (s.basak1@nuigalway.ie)

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224.

ABSTRACT Accurate head pose estimation from 2D image data is an essential component of applications such as driver monitoring systems, virtual reality technology, and human-computer interaction. It enables a better determination of user engagement and attentiveness. The most accurate head pose estimators are based on Deep Neural Networks that are trained with the supervised approach and rely primarily on the accuracy of training data. The acquisition of real head pose data with a wide variation of yaw, pitch and roll is a challenging task. Publicly available head pose datasets have limitations with respect to size, resolution, annotation accuracy and diversity. In this work, a methodology is proposed to generate pixel-perfect synthetic 2D headshot images rendered from high-quality 3D synthetic facial models with accurate head pose annotations. A diverse range of variations in age, race, and gender are also provided. The resulting dataset includes more than 300k pairs of RGB images with corresponding head pose annotations. A wide range of variations in pose, illumination and background are included. The dataset is evaluated by training a state-of-the-art head pose estimation model and testing against the popular evaluation-dataset Biwi. The results show that training with purely synthetic data generated using the proposed methodology achieves close to state-of-the-art results on head pose estimation which are originally trained on real human facial datasets. As there is a domain gap between the synthetic images and real-world images in the feature space, initial experimental results fall short of the current state-of-the-art. To reduce the domain gap, a semi-supervised visual domain adaptation approach is proposed, which simultaneously trains with the labelled synthetic data and the unlabeled real data. When domain adaptation is applied, a significant improvement in model performance is achieved. Additionally, by applying a data fusion-based transfer learning approach, better results are achieved than previously published work on this topic.

INDEX TERMS Head pose estimation, synthetic face, face dataset, visual domain adaptation.

I. INTRODUCTION

Head Pose Estimation (HPE) continues to be an active area of research in the computer vision (CV) domain because of its diverse application across a range of CV technologies. Highly accurate HPE is a key element for many next-generation consumer technologies which includes augmented and virtual reality (AR/VR) based entertainment systems, human-computer interaction technologies that engage human attentiveness and behaviour analysis, immersive audio systems

The associate editor coordinating the review of this manuscript and approving it for publication was Yongqiang Zhao¹.

and driver monitoring systems (DMS). In human behaviour analysis, HPE is used for estimating the human gaze and refining face analysis and authentication to infer the intentions, feelings, and desires of a user to personalize the associated system or technology to meet their needs. For DMS, HPE is important to monitor the driver's attention level. For AR/VR applications, HPE is used to predict the accurate field of view (FOV). HPE information is also useful in producing better face alignment for pose-robust facial authentication.

Head pose can be measured by the reading of sensors embedded in head-mounted-devices which are costly and awkward for users. Therefore, consumer-focused

technologies have increasingly adopted computer vision-based HPE that can estimate head pose with high accuracy and in real-time. Compared to wearable sensor-based methods, computer vision-based HPE is technically more challenging as it must handle variable factors such as facial expressions, occlusions, illumination conditions, and lens distortion in addition to the broad diversity of human facial appearance.

Computer-vision based HPE transforms the captured 2D facial images into directional data in three-dimensional space with three Euler angles: θ_x (Pitch), θ_y (Yaw) and θ_z (Roll). Figure 1 [1] shows the head model as a rotated object across the three different axes with the orientation of yaw, pitch and roll. Normally, the HPE algorithms follow two different approaches: geometry-based methods and learning-based methods. Geometry based methods take the key facial landmarks into consideration and estimate the pose through geometrical calculation. On the other hand, learning-based methods aim to extract features from the queried face images and predict the pose with the support of face datasets and their corresponding ground truth pose angles.

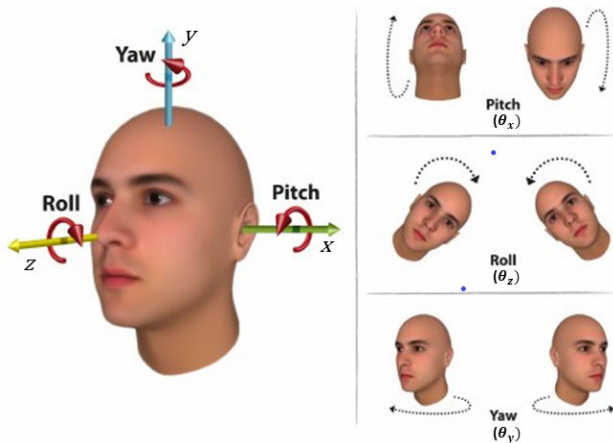


FIGURE 1. Head orientation with Pitch, Yaw and Roll [1].

These learning-based methods can be a regression or classification task. Regression approaches predict the head pose by fitting a regression model on the training data and estimate the yaw, pitch and roll in continuous angles, making these models comparatively complex. On the other hand, classification approaches mostly rely on putting the head pose into a discrete bin. These methods are comparatively robust to large pose variations but have a sparse solution space, e.g. 10 degrees intervals. for each bin.

Head pose estimation from a single image makes the problem more challenging. It requires learning the mapping between 2D and 3D spaces. Previously published works use different modalities like depth information [2]–[5], inertial measurement unit (IMU) [6] or video sequences [7] as a cue to map the features extracted from the 2D image to the 3D space. These methods require more computation and different sensors which are not always available. Therefore, because of its low computational cost and easy setup, HPE from a

single image makes is a popular area in HPE research. Most of these single image-based HPE methods ([8]–[10]) leverage the use of Convolution Neural Network (CNN) to extract features from the 2D images and use those high-level features to model 3D head pose regressors.

Though these Deep Neural Network (DNN) based methods have given good results, a major drawback of such supervised models is the requirement for accurately labelled data. Particularly for HPE tasks, it is challenging to obtain accurately annotated head pose data with variations of appearances like race, age, gender and other environmental factors like noise, illumination and occlusion.

Additionally, the acquisition of new data from human subjects now falls under different data protection and privacy regulations such as the General Data Protection Regulation (GDPR) and is subject to ethical review and increasingly stringent guidelines. Furthermore, some data acquisition measurements such as depth sensing and IMU motion are prone to sensor noise. Manually labelled key point approaches are also mostly giving inaccurate results because of unknown 3D models and camera parameters.

The head-pose datasets available captured from real subjects like Biwi Kinect Head Pose Dataset [2] and Pointing'04 [11] only comprise around 15k and 4k data samples from 20 and 14 subjects respectively. Among these two Biwi is most commonly used for benchmarking. But due to the limited size, neither of these datasets are suitable to train DNN based HPE models.

Generating synthetic facial images through Computer Graphics (CG) Software provides an inexpensive and sufficient amount of accurately labelled data with a comparatively low effort and complexity as the head models, camera parameters and positions, scene illuminations and other constraints can be controlled within the 3D environment.

Though this synthetic data can be perfectly annotated, training solely with the synthetic data can lead to outcomes that don't match the current state-of-the-art. It is hypothesized that this is due to the mismatch between the feature distribution of the synthetic (source) domain and the real-world images (target domain). This is known as the domain shift [12]. To address these challenges, there have been many recent studies on visual domain adaptation (DA) which is a particular variant of transfer learning. DA utilises the labelled data from a source domain and the unlabeled data from a target domain and learns how to reduce the gap between the two domains. In this work, a similar approach is used to learn the domain invariant features from the synthetic and real data and thus improve the model performance.

The main contributions of this work are as follows:

- A methodology to build a synthetic head pose dataset with the help of a commercially available 3D asset creation tool, iClone [13] and an open-source 3D computer graphics software, Blender [14].
- Using the proposed methodology, we propose a new synthetic head pose dataset with the corresponding ground truth head pose.

- Experimental results show that training a state-of-the-art HPE model solely with the new proposed dataset gives near state-of-the-art HPE result. Also, applying data-fusion-based transfer learning and fine-tuning the model with only 1k of real data is able to produce a better result than the previously published work.
- Finally, it is shown that by applying the visual adversarial domain adaptation technique and training the model with the labeled synthetic data and the unlabeled real data, it is able to learn domain invariant features and produce better results than training only with synthetic data.

The paper is structured in the following way – Section II reviews the recent work on HPE and visual DA along with the descriptions of the datasets available for the HPE task. Section III provides the foundation methods of head pose measurement in a 3D environment. Section IV and V describes the methodology of the synthetic data generation and dataset Details respectively. Section VI introduces the theory behind the Synthetic to Real Domain Adaptation. Section VII presents the model description and their implementation details along with the training strategy and experimental results. Finally, the paper concludes with a discussion on the results and conclusion with future work in section VIII and IX.

II. LITERATURE REVIEW

In this section, firstly, a review of recent research works and the current state-of-art in HPE methods is provided. Then, an overview of publicly available head pose datasets is presented, followed by the recent relevant works in visual domain adaptation.

A. HEAD POSE ESTIMATION METHODS

1) LEARNING FROM GEOMETRY

Geometry-based methods predict the head pose by geometrical calculation with the help of facial feature points. These methods take advantage of the geometric distribution of the facial key points from the 2D image. Initial work by Gee and Cipolla [15] considered the proportion between five facial key points and the length of the nose with a fixed value to calculate the head pose. Similarly, Nikolaidis and Pitas [16] used the isosceles triangle formed by the mouth and the two eyes to predict the yaw angle. To predict the yaw angle more accurately, Narayanan *et al.* [17] proposed a more generic geometric model with an ellipsoidal and cylindrical structure to customize 12 different head models. This only predicts the Yaw of the head. However, it is very difficult to estimate the head pose accurately with these fixed geometric models as the feature keypoint distributions of the human face vary a lot with race, age, genders like factors.

To overcome these challenges, another set of approaches have been proposed which aim to estimate the head-pose, mapping the facial key points from the 2D image to a 3D facial model. The head pose angles are then calculated from the elements of the rotation matrix which can be derived from

the projection mapping between the 2D face image and the 3D head model. The rotation matrix solution was first proposed by Fridman *et al.* [18] to estimate the head pose according to a 3D facial model and the corresponding 2D facial feature points directly.

A real-time 3D facial model had been used in previous work by Martin *et al.* [5] for the HPE task which introduced the iterative closest point algorithm (ICP) to find the best matching pair of the 2D facial image and the 3D head model. Meyer *et al.* [4] combined particle swarm optimization and the ICP algorithm to estimate the head pose. All the above methods used the depth cue of the facial image. In recent work, Yuan *et al.* [19] proposed a 3D morphing method with spherical parameterization which will deform an existing 3D facial model with the help of four non-coplanar 2D facial feature point along with all the three directions of yaw, pitch and roll.

2) LEARNING FROM FACIAL FEATURES

Learning-based methods are trained to find the relationship between the query images represented by the extracted appearance feature distributions along with the head positions and rotations. These methods are supported by a huge face training dataset annotated with the corresponding yaw, pitch and roll and uniformly distributed along with these label spaces.

These learning-based methods are mathematically formulated as a regression or classification problem to estimate the head pose from the features learnt from the 2D images. One of the initial works presented by Murphy-Chutorian and Trivedi [20] uses support vector regression and Localized Gradient Orientation histograms to predict the head orientation in a driver monitoring system. Ba and Odobez [21] improved the previous head tracking methods with Bayesian formulation by introducing a silhouette likelihood term with particle filtering.

A random forest model was used by Fanelli *et al.* [2] to estimate the head pose by learning the 2D features from the depth images. In this work, the leaf nodes with high training variance are filtered out. Tan *et al.* [22] extend the approach incorporating the 3D features and frame-by-frame temporal tracking through regression forest. The random forest-based method was further combined with Hough voting by Liang *et al.* [23] which varies the leaf weights with L0 regularization and prune the unreliable leaf nodes of the decision tree. Instead of segmenting the whole head, Riegler *et al.* [24] used a classifier to segment image patches into foreground and background and regression to cast vote in Hough space for the foreground patches. The approach is similar to Hough Forest but the Random Forest part was replaced with a Convolution Neural Network (CNN) and called it a Hough Network.

A transfer learning approach was used by Rajagopal *et al.* [25] which deals with the HPE as a classification problem from multi-view surveillance images with a small amount of target training data. Papazov *et al.* [26] proposed a novel approach to extract a triangular surface

patch (TSP) descriptor from a depth map and matched it with the pre-computed synthetic head models with a fast-nearest neighbour loop. The computed TSP is further used to estimate the 3D head pose and facial landmarks. A video sequence of synthetic facial images was used by Gu *et al.* [7] to learn the head pose and facial landmarks via temporal shift, though the video sequences require recurrent neural models with a high computational cost.

The above-mentioned methods mostly deal with the HPE as a classification task and used different modalities like facial depth as additional cues which are difficult to acquire. Therefore, deep learning-based HPE from a single facial RGB image without a facial landmark has gained interest among the research community in recent years. The initial work on this was proposed by Ahn *et al.* [27] which used CNN based models to regress the head pose information. Patacchiola and Cangelosi [28] examined adaptive gradient methods with different CNN architectures for HPE tasks. A ResNet based model was used by Chang *et al.* [29] to predict the head pose and facial key points jointly. To predict the head pose more accurately Ruiz *et al.* [8] used the ResNet50 backbone architecture and a multi-loss CNN (HopeNet) for feature extraction and combined loss stream of regression and binned pose classification. A lightweight structure FSA-Net for head pose feature regression, using the stage-wise regression model SSR-Net [30] was proposed by Yang *et al.* [9].

Few of the above works use augmented synthetic facial images with the ground truth head pose to train their models. Ruiz *et al.* [8] and Yang *et al.* [9] use the synthetically expanded dataset 300W-LP, which is created by augmenting real images. Gu *et al.* [7] introduced a synthetically created dataset SynHead, which has been rendered through a CG tool from a very high-quality 3D scan obtained from [31]. Wang *et al.* [32] also introduced a synthetically rendered head pose dataset from high-quality 3D scans and propose a fine to a coarse deep neural network to predict accurate head pose. However, the dataset is not publicly available for use. They use a transfer learning approach and train the network with a mix of synthetic data and real data which improves the model accuracy with better generalization. The model was trained with approximately 260k synthetic images from their dataset and 15k real images from the Biwi dataset.

B. AVAILABLE HEAD POSE DATASETS

There are few datasets available that have been used for monocular image-based HPE tasks.

1) 300W-LP & AFLW2000 3D

300W and AFLW2000 3D [33] databases were created and released at the same time. uses multiple alignment real face databases with 68 facial key points including LFPW, AFW, IBUG, HELEN and XM2VTS. These images are collected randomly from the web so there is no data available in terms of identity or the total number of subjects. It uses 3D Dense Face Alignment (3DDFA) in which a dense 3D Face model

is fitted to the images through a CNN and further synthesise robust profile views through a face profiling algorithm that align faces in large poses up to 90 degrees of yaw. The 300W database contains around 61225 samples with large poses, which is further expanded to 122450 samples by flipping. The combined dataset is called 300W across Large Pose (300W-LP). The AFLW2000-3D contains 2000 images in the wild.

2) AFLW

AFLW [34] contains 21080 real faces in-the-wild collected from the web with wide pose variations (yaw from -90 degree to $+90$ degree). The head poses are extracted with the help of the POSIT algorithm [35] and have been used for coarse HPE. But as the images are annotated with up to 21 visible landmarks the face alignments have errors and the model fitting accuracy is low [33].

3) BIWI

The Biwi Kinect Head Pose Dataset [2] contains approximately 15.7k images taken from 24 sequences of 20 subjects (8 women and 12 men, 4 people wearing glasses). The data was captured by a Kinect 1 depth sensor and the head orientation is labelled by a state-of-the-art template-based head tracker, where a generic template was deformed to match the specific subjects and the 3D head location and rotations were measured. Each sample has a resolution of 640×480 pixels with the faces containing 90×110 pixel on average. The head pose ranges from $\pm 75^\circ$ yaw, $\pm 60^\circ$ pitch and $\pm 50^\circ$ roll.

4) POINTING'04

Pointing'04 [11] has captured 2.7k images from 14 subjects. The head pose of the captured subjects is only represented by the two angles yaw and pitch and both have fixed interval of 15 degrees with 93 discrete poses. During the data acquisition, the subjects were asked to stare at different markers fixed in the room, which results in an error in the ground truth head pose values for many samples. The pre-trained model of the current state-of-the-art HPE FSA-Net gives a Mean Absolute Error [MAE] of around 10 degrees when tested on this dataset.

5) BOSPHORUS

The Bosphorus [36] dataset is captured by using a 3D structured light system that contains 4666 images with 13 systematic head poses. To give the Yaw rotation subjects were asked to align themselves in a rotating chair, while for the pitch, subjects were required to look at the marks on the wall. Because of the data accusation method, the ground truth pose angles are prone to error. The dataset contains seven yaw angles, four-pitch and two cross rotations. Apart from the pose annotations it also has a variety of facial expressions and occlusions like hand, hair and eyeglasses.

6) SASE

The SASE dataset [37] has captured different head poses from 50 subjects (32 males and 18 females) via the Kinect 2.

TABLE 1. A comparison of different head pose datasets.

Database	Samples	Aquisition	Subjects	Facial Landmarks	Pose Descriptions	Released
Pointing'04 [11]	2790	Lab	14 Real		Discrete Yaw: $[-90^\circ, -75^\circ, -60^\circ, -45^\circ, -30^\circ, -15^\circ, 0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ]$, Pitch: $[-90^\circ, -60^\circ, -30^\circ, -15^\circ, 0^\circ, 15^\circ, 30^\circ, 60^\circ, 90^\circ]$	2004
Bosphorus [36]	4652	Lab	105 Real	24	Discrete Yaw: $[-90^\circ, -45^\circ, 10^\circ, 20^\circ, 30^\circ, 45^\circ, 90^\circ]$ Cross rotations: $[(45^\circ \text{ yaw}, -20^\circ \text{ pitch}), (45^\circ \text{ yaw}, 20^\circ \text{ pitch})]$ Pitch: slight upwards, slight downwards, downwards, right-downwards, right-upwards	2008
Biwi Kinect [2]	15 K	Lab	20 Real	-	Continuous Yaw: from -75° to $+75^\circ$; Pitch: from -60° to 60° ; Roll: from -50° to 50°	2013
SASE [37]	30 K	Lab	50 Real	-	Continuous Yaw: from -75° to $+75^\circ$; Pitch: from -45° to 45° ; Roll: from -45° to 45°	2016
AFLW [34]	25993	Web	Random Collected from web	21 visible landmarks	Continuous Annotated by algorithm on 21 landmarks leading to erroneous pose	2011
300W-LP [33]	122450	Web	Random Collected from web	68	Continuous Annotated by algorithm on 68 landmarks	2016
AFLW2000-3D [33]	2000	Web	First 2000 sample from AFLW	68	Continuous Annotated by algorithm on 68 landmarks	2016
SynHead [7]	510960	Synthetic Rendered	10 Synthetic Head	-	Continuous Followed the Biwi sequence	2017

Altogether the dataset consists of around 30k images with 600+ frames per subject. The head orientation has been obtained by calculating the positions of five markers stuck on each participant's face and deriving the rotation matrix between the initial and current vectors.

7) SynHead

NVIDIA SynHead [7] contains 510960 frames of 70 head motion tracker rendered using 10 individual high-quality 3D scan head models from [31]. It contains head motion tracks of all 24 Biwi sequences, though it was rendered with a different sequence of the rotation from that was followed by Biwi.

A comparison of the different features of these databases is shown in Table 1. Out of these datasets, because of their limitations of size, only the 300W-LP dataset is suitable for DNN training. Even though the SynHead dataset has a large number of synthetic head pose frames, it only contains 10 individual subjects from high-quality 3D scans, which make it less diverse and expensive to acquire. On the contrary, the dataset produced in this work has more than 300k frames from 100 individual models.

C. VISUAL DOMAIN ADAPTATION

Visual domain adaptation (DA) tries to learn the domain invariant features when there is a gap between the feature distribution of the source data on which the network is being trained and the target data on which the network is to be evaluated. It tries to reduce the gap between these two domain distributions. Almost all of the previous work on DA has been

proposed on classification tasks where the data distribution has shared label spaces, in other words, the source and the target data have a similar set of class labels. However, for regression problems, this scenario is not valid as it has a continuous label distribution.

The earliest and most prominent work on DA was proposed by Ganin and Lempitsky [38] with the domain adversarial neural network (DANN) which assumes identical labels spaces where for every sample of the source data there exists a target data with the same label class. However, in the real world, this assumption does not stand as only a small amount of target domain data exists. Therefore, while training the DANN in such a scenario both source and target labels are aligned with each other but as the target label space is not matched with the source labels it causes negative transfer. To solve this issue Cao *et al.* [39] introduced partial adversarial domain adaptation (PADA) which tries to reduce the negative transfer due to a mismatch between source and target domain labels by downweighing the source class data which has a low probability of existence in the target data.

There are many subsequent works [40], [41] that refine PADA by eliminating the source samples which are not present in target data through different weighting schemes. But all these approaches work on classification tasks where they consider partially shared label spaces. For HPE the label space is a continuous distribution, so these proposed methods cannot be applied directly to the HPE problem. The only work that deals with domain adaptation on the regression task, specifically on HPE, is proposed by Kuhnke and Ostermann [42], which reduces the negative

transfer from the source outliers through generating source sampler weights during training and propose Partial Adversarial Domain Adaptation for Continuous label spaces (PADACO). This is the only work that trains only on synthetic data rendered from a CG tool and tests on real data. In this article, a similar but relatively straightforward sampling strategy has been used to obtain data samples from the source domain thus reducing negative transfer during adversarial training.

III. HEAD POSE REPRESENTATION WITH 3D GEOMETRY

In this section, the 3D representation of the head pose is discussed. As the head is rotated along with the X, Y and Z axis, the head pose can be represented with the corresponding Euler angles θ_x (Pitch), θ_y (Yaw) and θ_z (Roll) as shown in figure 1.

When a point at (x, y, z) in 3D world coordinates is rotated around the X-axis with an angle of θ_x the new co-ordinate of the point will be –

$$(x_x y_x z_x) = R_x \cdot (xyz)^T \tag{1}$$

where

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

In the same way, if the point rotates around Y and Z axis with an angle of θ_y and θ_z respectively the modified coordinates of the point will be –

$$(x_y y_y z_y) = R_y \cdot (x y z)^T \tag{3}$$

and

$$(x_z y_z z_z) = R_z \cdot (x y z)^T \tag{4}$$

where

$$R_y = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

and

$$R_z = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 & 0 \\ \sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

So, combining (2, 5, 6) for a rotation of a point along all the axes, the final coordinates of the point will be –

$$(x_{xyz} y_{xyz} z_{xyz})^T = R_x R_y R_z \cdot (x y z)^T = R \cdot (x y z)^T \tag{7}$$

where,

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

R is known as the rotation matrix and the Euler angles θ_x , θ_y and θ_z can be calculated as –

$$\begin{cases} \theta_x = \tan^{-1} \frac{r_{32}}{r_{33}} \\ \theta_y = -\tan^{-1} \frac{r_{31}}{\sqrt{r_{32}^2 + r_{33}^2}} \\ \theta_z = \tan^{-1} \frac{r_{21}}{r_{11}} \end{cases} \tag{9}$$

Additionally, the translation of any point in 3D space is provided by the translation matrix as –

$$T(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

where d_x, d_y, d_z are the displacement of any point along the x, y, z-axis respectively.

Blender provides the transformation matrix combining the three rotation and translation matrix as $TR_x R_y R_z$, so the individual Euler rotation of yaw, pitch and roll can be calculated with equation 9.

IV. DATA GENERATION METHODOLOGY

In this section, the detailed methodology of creating a synthetic dataset is discussed. As outlined in section II-B of the literature review most of the datasets currently available for head pose estimation have a very limited amount of ground truth image and label pairs which makes them unsuitable for training deep learning models. Also, due to practical limitations in data acquisition, most of the datasets' ground truths are prone to errors, especially in high concatenated-rotation (combination of yaw, pitch and roll or combination of any two) angles. Therefore, as an alternative to the real data, this work presents this methodology using a commercially available 3D asset creation software and an opensource 3D CG tool to generate synthetic facial images along with the ground truth head pose.

A. 3D SCENE SETUP WITH VIRTUAL HUMAN MODELS

Previous works [7], [32], [42] with synthetic virtual humans mostly used high-quality 3D scans to generate synthetic data from 3D human models. But these 3D scans are expensive and difficult to capture due to different data regulation laws like GDPR, so there is a very limited number of variations in the currently available synthetic head pose data. As an alternative to generating the virtual human models, this work uses the low-cost commercially available software iClone 7 and Character Creator [43]. The Character Creator comes with a ‘‘Realistic Human 100’’ package consisting of 100 human models of different age, race, gender, thus reducing the bias of the dataset. A sample of these models can be found in figure 2. The iClone tools also provide a feature to add different facial expressions and the facial morph can also be changed to add variation in the 3D mesh as shown in figure 3.

As iClone cannot capture ground truth like facial depth, head pose, camera location, scene illumination all the models

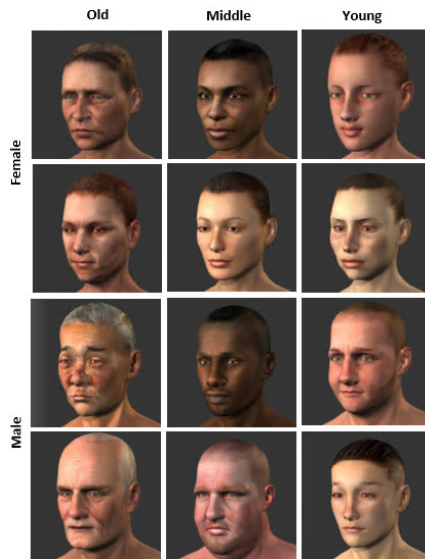


FIGURE 2. Samples from the 100 Realistic Head Models with variation in gender, race and age.

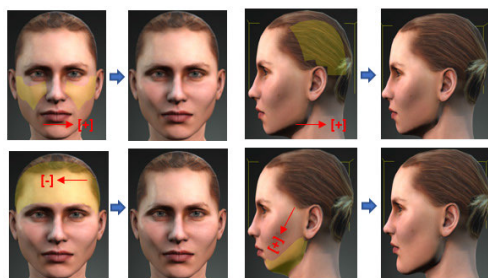


FIGURE 3. Applying change in the morph to add variations in the head models in iClone [45].

need to be exported for further data capture. The models can be exported in the commonly supported format by any 3D modelling software including alembic, FBX and obj. In this work, all models are exported from iClone in FBX format with Physically Based Rendering textures (Metallic, Diffuse, Roughness, Opacity) to add realism.

These fully rigged models in FBX format are then imported into Blender [14]. Blender is an opensource computer graphics (CG) software with Python integration. To animate the rigs, keyframes can be added with constraints and shape keys commonly known as morph targets or blend shapes. Also, the camera can be added to the scene which comes with properties like FOV, a camera near and far clip value, sensor size, depth of field and f-stop value which help to replicate a real-world camera configuration. It also comes with the realistic Cycle rendering engine which uses path tracing [44]. Path tracing tracks the path of light and considers refraction, reflection and absorption to make the rendering realistic. The full-featured workflow used in Blender is shown in figure 4. The FBX models exported from iClone contain the fully rigged armature with the mesh which can be used to add motions to the head.

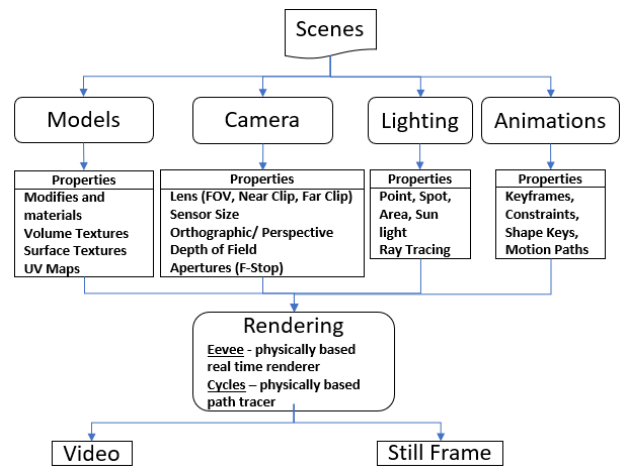


FIGURE 4. Workflow and different features of Blender [45].

A sample model is shown in figure 5. To vary the scene light, different illuminations available in Blender were used including area, sun, point, and spotlight. To render the ground truth image, a camera model has been added to the scene in perspective mode with the Cycle rendering engine selected. The detailed methodology can be found in [45]. To add variations to the background, a combination of plain, textured, and real images have been chosen.

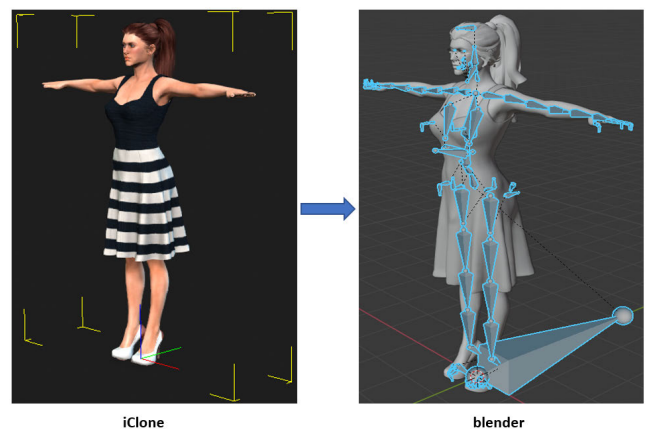


FIGURE 5. Importing the fully rigged FBX models from iClone to Blender [45].

B. APPLYING HEAD POSE TO 3D HUMAN MODELS

To generate the ground truth data, a sequence of head movements need to be applied to the FBX models. As these models are fully rigged, the neck bone is selected to provide the rotation to the head mesh. An empty object has been added to the centre of the two eyeballs which has been chosen as the centre of the head and the camera optical axis will be normal to this point to ensure the initial head position. Figure 6 shows the neck bone and the empty axis object highlighted. The translation and the rotation of the neck bone have been copied to the empty object which constraint the empty object to follow the neck bone.

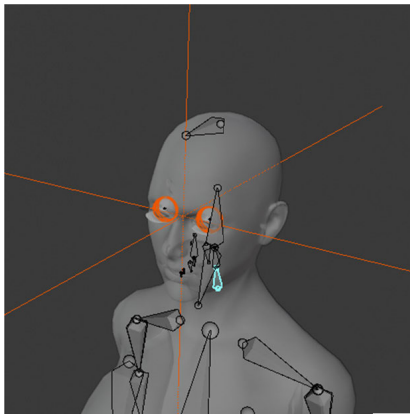


FIGURE 6. Neck bone highlighted in cyan on which the head rotation has been applied and the empty object at the center of the two eyeballs highlighted in orange.

As the head movement cannot be controlled mathematically in iClone when the default models are imported in Blender, the head is not at its zero position (yaw, pitch and roll at 0°). To set the initial frame of the head where the yaw, pitch and roll of the head are zero along with the Blender world co-ordinate, the main neck bone was rotated in such a way that the rotation of the empty object in blender local co-ordinate becomes zero along the x, y and z-axis. This has been achieved iteratively through a Python script minimizing the delta of the rotation of the empty axis along with the three-axis.

After the initial setup, uniform rotations have been applied to the neck bone in the sequence of PRY (pitch, roll and yaw) and all the frames have been saved. Blender provides the rotation matrix for the empty object from which the exact head pose in yaw, pitch and roll have been calculated with the help of equation 9. A sample of applying the head pose is shown in figure 7. Following most of the previous datasets' range the yaw, pitch and roll have been varied in the range of $\pm 80^\circ$, $\pm 70^\circ$ and $\pm 55^\circ$, respectively in an interval of 3° .

Though these rotations cover a wide range of angles, as these are linear sequences, some of the cross-rotation angles are not covered. As in Biwi the head pose angles are captured tracking the real human subjects the ground truth head pose sequences of the Biwi database has been collected and applied to the head models similar to SynHead [7]. This will also help to compare the evaluation result with the Biwi dataset later. The head mesh vertices have different weights with respect to the neck bone, so the rotation values of the empty axis object and the neck bone are not equal. Also, the 100 head models are rigged differently with different mesh weights so the transformation relation between the neck bone and the empty axis object is different for each of these models. The transformation between these two objects for all the 100 realistic virtual humans has been learnt individually by training a shallow fully connected neural network from the data collected in the previous step where a uniform rotation has been given to the neck bone. After applying these learnt models, the actual rotation of the neck bone for each Biwi

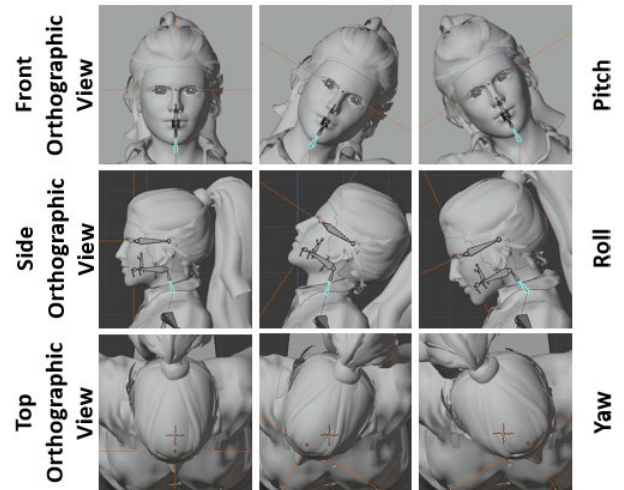


FIGURE 7. Applying head pose along the three axes with respect to the neck bone highlighted.

ground truth sequence is calculated so that the rotation of the empty axis matches with the Biwi sequences. After applying the Euler angles learnt from 24 Biwi sequences, all the frames have been recorded. However, as the rotations were applied to the internal neck bone, the head mesh was not exactly aligned with the Biwi sequences. The mean average error with Biwi for these sequences is approx. 1° in Euler scale.

C. GENERATING GROUND TRUTHS

To collect the ground truth, the camera added to the scene was set up in such a way that the camera optical axis is aligned with the empty object axis as stated in the previous step. The camera is set at a distance of 30 centimetres from the nose tip of the model and the background plane is at a distance of 2 meters. Therefore, to cover the whole scene the near and far clip of the camera is set to 0.001 and 5.0 meters, respectively. The camera sensor size and field of view (FOV) are set at 36 millimetres and 60° . To obtain the final render, the RGB render pass was used in the Blender compositor setup. As stated in the previous section, the background of the scene was varied to provide more variations in order to improve model generalization. For the textured background, the Brodatz-based colour images provided by Abdelmounaime and Dong-Chen [46] are used. For the real background, the images provided by the SynHead [7] dataset in the background folder are selected.

The rotations recorded in the previous step are applied to the model and the corresponding frames are rendered. For each frame, the current translation and rotation (in Euler) of the empty object has been captured through an automated python script in Blender world co-ordinate. The rendering of ground truth is carried out in an Intel Core i5-7400 3 GHz CPU machine with 32 GB of RAM and an NVIDIA GeForce GTX TITAN X Graphical Processing Unit (GPU) with 12 GB of dedicated graphics memory. The RGB ground truth head pose images are rendered from the 3D model with a resolution



FIGURE 8. Samples from the generated synthetic data with different variation of head pose. The first three rows show the data with a plain background, the fourth and fifth rows show data with textured backgrounds and the last two row shows data with real backgrounds.

of 640×480 pixels in jpeg format. Each 2D image frame took 26.3 seconds on average to render using *Cycle Rendering Engine* which is Blender's physically-based path tracer for production rendering.

V. DATASET DETAILS

Following the above-discussed methodology, the ground truth RGB images and their corresponding ground truth models for 44 female and 56 male models have been generated. As ground truth, different attributes like camera initial location, camera initial rotation, camera post location, camera post-rotation have been collected when the camera location has been varied. Additionally, the initial location and rotation of the empty object and the post-rotation and location of the same has also been captured and saved in a text file for each frame. Each subject has approx. 3.5k 2D image samples which make the total dataset size to around 3,500k image samples. The data is stored in an individual folder for the 100 head models. For each head model folder, the rendered images and corresponding ground truth are stored in three different paths for the three type of backgrounds – simple, textured, and real. The zipped version of the total dataset consumes around 60 GB of disk space. A sample of images from the generated data with varying Pitch, Yaw and Roll has been shown in figure 8. The dataset will be released and can be accessed through the GitHub page.¹ While training a deep neural network, the generalization of the model is highly dependent on the statistical data distribution of the dataset. Thus, to check the label distribution, several identities from the dataset has been selected and label distributions are compared with those from the Biwi dataset. Figure 9 shows

the two distributions which show the generated dataset is more uniform across the value of yaw, pitch, and roll, whereas the distribution of Biwi shows it is mainly concentrated on the angles near the centre.

VI. SYNTHETIC TO REAL DOMAIN ADAPTATION

As stated in the introduction section, this synthetic data is annotated perfectly without any error, but training any deep learning model solely with synthetic data can lead to the poor performance of the models because of the domain mismatch between synthetic and real. Therefore, the visual domain adaptation will help to reduce the feature gap between synthetic and real domain data. In this section, the theory and the common notation behind the domain adaptation will be explained.

In any machine learning task, a domain D is made up of a feature space X with a probability distribution $P(X)$ where $X = \{x_1, \dots, x_n\}$. For a specific domain, $D = \{X, P(X)\}$ a machine learning task T is trying to learn the objective function $f(\cdot)$ from a feature space Y , which in another way can be a probability distribution $P(Y|X)$. In general, this $P(Y|X)$ can be learnt from the labelled data $\{x_i, y_i\}$ where $x_i \in X$ and $y_i \in Y$.

However, a typical domain adaptation (DA) task consists of two domains: a source domain $D^S = \{X^S, P(X)^S\}$ with the corresponding label $y_i \in Y_S$ and a target domain with no labelled data $D^T = \{X^T, P(X)^T\}$. In this work, the source domain data is the synthetic head pose data with the ground truth head pose and the target domain is the real head images where there is no labelled head pose associated with these images. In traditional DA a common assumption is that the source domain label space C_S and the target label space C_T are shared. In partial domain adaptation (PDA) the target label

¹<https://github.com/C3Imaging/SyntheticHeadPose>

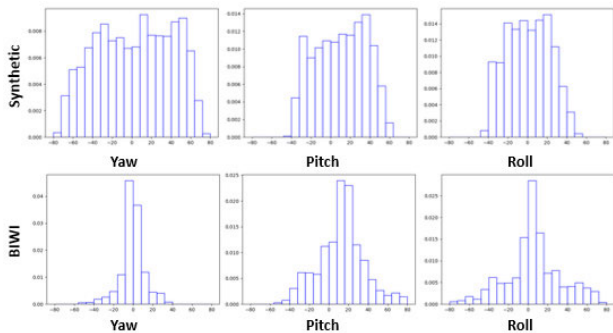


FIGURE 9. The first row shows the label distribution of the generated data across yaw, pitch and roll. The second row shows the similar distribution of Biwi data.

space C_T is a subset of the source domain label space C_S , and the rest of the labels in the source domain are seen as outliers. As the DANN tries to align the source and target distribution it will also align the label simultaneously. However, as there are outliers in the target distribution, this causes negative transfer during training. PADA overcomes these challenges by down-weighting the contribution of the source data which has a lower probability of existence in the target distribution. This methodology works well for classification tasks as the labels are fixed. But the same strategy cannot be applied to a regression task (i.e. head pose estimation), as it has a continuous label space. Therefore in this work, similar to [42] the source data with the nearest match with the target predicted distribution has been sampled during the Domain Adaptation training phase.

A basic DANN [38] normally has three subnetworks: A feature extractor G_F , which learn the feature from the input images, a network for the actual task, in this case, the head pose regressor G_Y which regress the actual head pose from the input image, and a domain classifier G_D , which is trained to differentiate the target domain from the source domain. The main goal of the DA is to match the feature distribution of the source and the target domain is achieved by a two-player minimax game between G_D and G_F which tries to confuse G_D to learn the indistinguishable features from the source and target domain.

To achieve the minimax goal during the training phase, the parameters θ_D of the domain classifier G_D are learnt by minimizing the cross-entropy loss of G_D , at the same time the parameters θ_F of the feature extractor G_F tries to maximise the loss G_D to confuse it. Simultaneously the pose regressor G_Y is trained to learn the parameters θ_Y for the actual task, in this case, the head pose estimation. So the overall objective function can be expressed as –

$$J(\theta_F, \theta_Y, \theta_D) = L_Y \left(G_Y \left(G_F \left(x_i^S \right) \right), y_i \right) - \mu L_D \times \left(G_D \left(G_F \left(x_i^S \cup x_i^T \right) \right), l_i^S \cup l_i^T \right) \quad (11)$$

where L_Y is the main task loss (pose regressor loss) and L_D is the domain classifier loss. μ is the hyperparameter to make a trade-off between L_Y and L_D . To train the domain discriminator as a binary classifier, the source and target

domain data are labelled as 1 and 0 respectively which are denoted as l_i^S and l_i^T in Eq. (11).

To obtain the desired saddle point of Eq. (11) in the minimax optimization of the parameters of the network $(\hat{\theta}_F, \hat{\theta}_Y, \hat{\theta}_D)$ is learned by converging –

$$\begin{aligned} (\hat{\theta}_F, \hat{\theta}_Y) &= \arg \min_{\theta_F, \theta_Y} J(\theta_F, \theta_Y, \theta_D), \\ (\hat{\theta}_D) &= \arg \min_{\theta_D} J(\theta_F, \theta_Y, \theta_D) \end{aligned} \quad (12)$$

The minimax optimization can be achieved through iterative training using Generative Adversarial Networks (GAN) [47] or the Gradient Reversal Layer (GRL) proposed in Ganin and Lempitsky [38]. In this work, the GRL approach has been used. The GRL has no trainable parameters except for the hyperparameter μ . During the training of the network, GRL produces an identity transform in the forward pass and during backpropagation GRL takes the gradients from the previous layer multiplied with the negative weight $-\mu$, and pass them to the preceding layer. This GRL layer is inserted between the feature extractor G_F and the domain classifier G_D . So effectively the partial derivative of the loss $\frac{\partial L_D}{\partial \theta_F}$ is replaced by $-\mu \frac{\partial L_D}{\partial \theta_F}$ which helps to reach the saddle point during the minimax optimization.

VII. EVALUATION OF THE DATA

In this section, first, the details of the state-of-the-art model that is used in this work to evaluate the effectiveness of the generated synthetic data are discussed including the domain adaptation module that is added to the existing model architecture. Next, the training strategy is presented, followed by the experimental details and results.

A. DETAILS OF THE MODEL

To evaluate how useful the generated synthetic data is for training HPE models, a recent state-of-the-art model FSA-Net [9] is selected. In its original work, this model has been trained on 300W-LP and Biwi and been validated against Biwi. The FSA-Net model is based on feature aggregation and a soft stagewise regression introduced in the work of SSR-Net [30] which employs a coarse-to-fine strategy for classification following the stage-wise regression. The soft stagewise regression (SSR) function accepts N set of stage parameters $\{\vec{p}^{(n)}, \vec{\eta}^{(n)}, \Delta_n\}$.

1) FEATURE AGGREGATION MODULE

FSA-Net employs a spatial grouping of features and passes it to the aggregation module. The feature map U_n for the n^{th} stage is a spatial grid that contains a k dimensional feature representation of a particular spatial location. Then to extract the pixel-level feature it computes an attention map A_n through a scoring function. The original work was based on three different scoring options (1) Uniform, (2) 1×1 convolution and (3) Variance. In this work the third strategy is used, in which the features are selected through

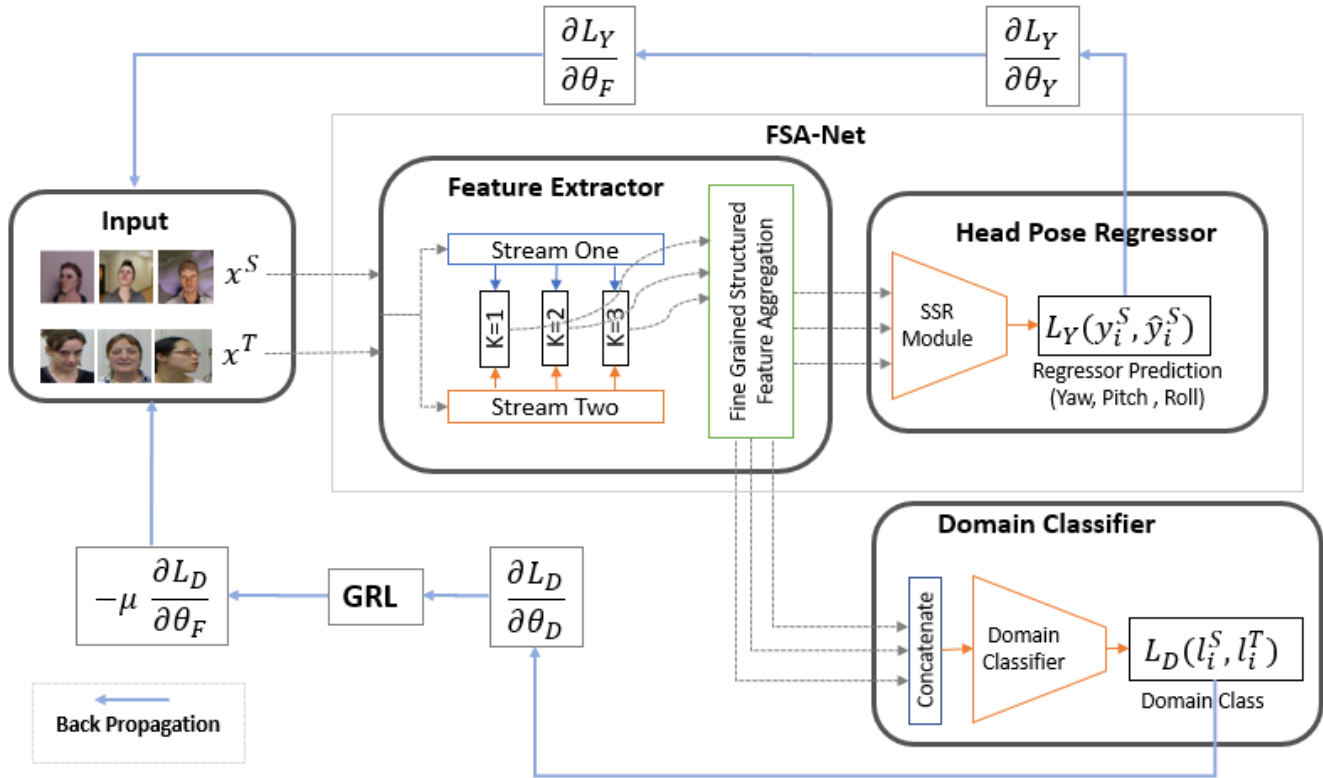


FIGURE 10. FSA-Net with the Domain Classifier and GRL layer for the adversarial learning.

Variance, which is differentiable but not learnable and comparatively less complex. After getting the feature map U_n and attention map A_n , a set of representative features \tilde{U}_n is extracted through $\tilde{U}_n = S_n U_n$. S_n is a linear dimensionality reduction transformation that has been learned from the attention map A_n . This representative feature \tilde{U}_n is then fed to the existing feature aggregation method capsule [48] to get the representative features V .

2) SSR-NET MODULE

The SSR-Net employs a coarse-to-fine architecture for classification following the soft stage wise regression. The classification divides the task into several bins of head pose (yaw, pitch and roll). A scale factor Δ_n defines the width of the bin and a shift vector $\vec{\eta}^{(n)}$ predict the center of each bin. The SSR soft stagewise regression function takes N sets of stage parameters $\{\vec{p}^{(n)}, \vec{\eta}^{(n)}, \Delta_n\}$ as input, where $\vec{p}^{(n)}$ is the probability distribution of the n th stage. These stage parameters are obtained from the final set of feature vector V of the feature aggregation module. The final regressor output of the head pose then thus obtained by

$$\check{y} = \sum_{n=1}^N \vec{p}^{(n)} \cdot \vec{\mu}^{(n)} \quad (13)$$

where $\vec{\mu}^{(n)}$ is a vector for representative values of head pose group and obtained from $\vec{\eta}^{(n)}$ and Δ_n .

3) DOMAIN ADAPTATION MODULE

To apply the domain adaptation technique during the training phase a domain classifier and the GRL layer have been

added to the existing FSA-Net model. A very shallow fully connected binary classifier network comprising of (Linear \rightarrow BatchNorm \rightarrow Linear \rightarrow ReLU \rightarrow Linear) has been designed for the domain classification task. The fine-grained feature stream from the FSA-Net feature aggregation layer has been concatenated and send to the domain classifier layer. The GRL layer has been injected between the feature aggregation and the domain classifier layer to produce the minimax optimization. The classifier and the GRL layer helps the adversarial learning during backpropagation. The overall model architecture is shown in figure 10.

4) LOSS FUNCTION

The end goal of the HPE task is to learn a representative function $F(x)$ which predicts the head pose \check{y} for an input image x . To find $F(x)$ the most common loss function found in HPE literature, the mean absolute error (MAE) between the ground truth and predicted head poses has been used here

$$L(y, \check{y}) = \frac{1}{M} \sum_{m=1}^M \|\check{y}_m - y_m\| \quad (14)$$

where y_m is the corresponding ground truth and $\check{y}_m = F(x_m)$ is the predicted pose for the image x_m .

For the domain classifier, the common cross-entropy loss has been used –

$$L_{\text{cross-entropy}}(y, \hat{y}) = - \sum_i y_i \log(\hat{y}_i) \quad (15)$$

where y is the true label distribution and \hat{y} is the predicted label distribution.

B. TRAINING METHODOLOGY

The FSA-Net fine-grained feature aggregation learns the feature from the training images from both source synthetic domain and target real images. The SSR-Net regression module helps to learn the head pose estimation task. The adversarial learning of the domain invariant features from the source and target domain is achieved by training the domain classifier and passing the backpropagation through the gradient reversal layer. During this adversarial training to reduce the negative transfer due to label mismatch from the source to target domain data a similar strategy to the work of Kuhnke and Ostermann [42] has been used to sample out the nearest source samples in terms of head pose from the target data. The overall training strategy is as follows –

- Inputs – Source Domain Synthetic images X^S with ground truth head pose Y^S , and target domain real images X^T without any ground, truth head pose labels.
- Step 1 – Divide the training source domain data into two sets. Train the FSA-Net which comprises of the feature extractor G_F and the head pose regressor G_Y with only the first set of source domain data (X^S, Y^S) to learn the parameters $\hat{\theta}_F$ and $\hat{\theta}_Y$ respectively and save the best model.
- Step 2 – Predict the head pose for each sample from the target domain with the model learnt from step 1 as $\hat{y}_i^t \leftarrow G_Y(G_F(x_i^t))$. Extract the nearest sample (image and ground truth label pairs) from the second set of source domain data for each target set image. The nearest neighbour sample is identified by the shortest distance calculated with the mean square error between the ground truth values from the source domain data and the predicted label \hat{y}_i^t from the target domain.
- Step 3 – After extracting the nearest samples from the source domain data the feature extractor G_F , head pose regressor G_Y and the domain classifier G_D are trained simultaneously with both source and target domain data. G_Y is trained with the sampled source domain data (X^S, Y^S), G_D is trained through adversarial learning with the source and target data (X^S, X^T) and their corresponding labels (l^S, l^T). Finally the respective parameters $\hat{\theta}_F$, $\hat{\theta}_Y$ and $\hat{\theta}_D$ are learnt.

C. EXPERIMENTAL DETAILS & RESULTS

Before running any experiments, the data is prepared by processing all the generated synthetic images through a popular face detector MTCNN [49] to loosely crop the face. To evaluate the data and to check if the data generated by the methodology mentioned in this work is close enough to the real-world data three different sets of experiments have been carried out on the dataset. All the experiments have been performed in an Intel I7 CPU and an Nvidia TITAN X GPU.

1) TRAIN ON SYNTHETIC DATA WITHOUT ANY TRANSFER LEARNING OR DATA AUGMENTATION

First, the original FSA-Net model is trained without any domain adaptation module and transfer learning methods

(i.e. only with the generated synthetic data) and tested on the two real datasets Biwi and SASE.

To replicate the real-world data, random Gaussian noise is added to the synthetic images during training, but no further data augmentation strategy is applied. The training set consists of 300k labelled synthetic images. The model is trained for 90 epochs with the Adam optimizer. The initial learning rate has been set to 0.0001, later the learning rate has been reduced gradually after every 30 epochs by a factor of 0.1.

There is no previous work published that deals with the HPE task training only on synthetic data and evaluating it with real data. The nearest scenario can be training the network with the synthesised 300W-LP data which was produced by augmenting the real data as discussed in section II-B and validating the trained model on the Biwi dataset which is a real dataset. Therefore, the results of the trained model are compared against this scenario. Also, as the only true synthetic data with head pose annotation that is currently available is SynHead, the same FSA-Net model has been trained with SynHead and has been evaluated against Biwi.

Table 2 shows the results of these scenarios. It includes three state-of-the-art HPE models that are all trained on the 300W-LP dataset and tested on Biwi. FAN [50] is a landmark detection method that produces multi-scale information and merged the block features. The accurate head pose then can be calculated from the detected landmarks. Hopenet [8] and FSA-Net [9] are landmark free regression methods for HPE task. The result shows training the FSA-Net with the synthetic data generated from this work reaches near the state-of-the-art results and perform quite well compared to the available Synhead dataset. It is also able to beat the landmark-based FAN result by more than 1° in MAE.

To analyse further and to understand the performance of the trained model on particular head pose angles both the FSA-Net models trained on the synthetic data produced by this work and Synhead are evaluated against Biwi in narrower angle ranges. Table 3 shows the result filtered yaw, pitch and roll (stated as Y, P and R respectively) from Biwi. It can be found that training solely with the synthetic data produced by this work can reach the state-of-the-art result in most of the narrow-filtered head pose angles. Also, it produces a better result compared to the Synhead dataset.

2) TRANSFER LEARNING WITH DATA FUSION

In the second phase of the experiments, a data fusion based transfer learning approach is applied during training where the FSA-Net model is first trained with the synthetic data and then the model is fine-tuned on a small set of real data from Biwi and SASE. In this experiment, the FSA-Net model is trained with around 70k of synthetic data and then the trained model is fine-tuned with around 1k of Biwi data. A similar experiment is conducted with SASE data as well.

The only similar work was done by Wang *et al.* [32] where 260k synthetic images and 15k of real images have been used. Both the real and synthetic images were split into 80% for training and 20% for testing. Experimental results are shown

TABLE 2. Experimental result – a comparison with recent research works with FSA-Net trained with the synthetic data.

Model	Training Set	Test Set	MAE	Yaw	Pitch	Roll
FAN [48]	300W-LP	Biwi	7.89	8.53	7.48	7.63
Hopenet [8]	300W-LP	Biwi	4.90	4.81	6.61	3.27
FSA-Net Fusion Capsule [9]	300W-LP	Biwi	4.28	4.56	5.21	3.07
	300W-LP	SASE	5.59	5.77	7.27	3.72
	Our Synthetic Data	Biwi	6.34	5.86	6.51	6.63
		SASE	6.63	6.52	7.76	5.61
	SynHead	Biwi	8.29	6.04	8.58	9.82

TABLE 3. Comparative evaluation of our data against the synhead dataset on the fsa-net model without any domain adaptation and training only on synthetic data and testing on Biwi varying the head pose along with one or two axis.

Range	Training Dataset	MAE	Yaw	Pitch	Roll
Y($\pm 90^\circ$), P($\pm 10^\circ$), R($\pm 10^\circ$)	SynHead	5.431	4.241	7.766	4.288
	Ours	3.324	4.025	3.433	2.516
Y($\pm 10^\circ$), P($\pm 90^\circ$), R($\pm 10^\circ$)	SynHead	4.408	4.681	6.144	2.400
	Ours	3.300	3.764	3.955	2.180
Y($\pm 10^\circ$), P($\pm 10^\circ$), R($\pm 90^\circ$)	SynHead	4.151	3.892	6.188	2.373
	Ours	3.091	3.587	3.690	1.998
Y($\pm 90^\circ$), P($\pm 90^\circ$), R($\pm 10^\circ$)	SynHead	6.203	4.972	7.196	6.441
	Ours	4.413	4.442	4.870	3.926
Y($\pm 10^\circ$), P($\pm 90^\circ$), R($\pm 90^\circ$)	SynHead	4.796	4.870	6.407	3.111
	Ours	3.755	4.515	4.130	2.621
Y($\pm 90^\circ$), P($\pm 10^\circ$), R($\pm 90^\circ$)	SynHead	5.722	4.439	8.228	4.497
	Ours	3.608	4.377	3.619	2.828

in Table 4 that include the results from this work and the related previous work [32]. It shows that fine-tuning the pre-trained model (trained only with synthetic data) with only 1k of the real image and ground truth pairs from Biwi can beat the previous work.

3) TRANSFER LEARNING WITH DOMAIN ADAPTATION (SEMI-SUPERVISED APPROACH)

In the third and final experiment, the domain adaptation approach with the training strategy discussed previously in section VII-B was used. The FSA-Net model is first trained with only the synthetic data for 70 epochs and the best model is selected by testing on a held-out test set from the synthetic dataset. Then the trained model is used to predict the pose of the real data sequences from Biwi and with the predicted result the nearest data is sampled from the synthetic data for every sequence of real data. Afterwards, the FSA-Net with the domain adaptation module is trained using those sampled synthetic data and real data for another 30 epochs. In this phase of the experiment both the real (Biwi) and the

TABLE 4. Mean error of yaw, pitch and roll on transfer learning approach with data fusion.

Model	Training Set	Test Set	Yaw	Pitch	Roll
Wang [32]	Synthetic (208k) + Biwi(12k)	Biwi (3k)	4.76	5.48	4.29
Fsa-Net [9]	Our Synthetic (300k) + Biwi(1k)	Biwi (14k)	4.620	4.537	3.33
Fsa-Net [9]	Our Synthetic + SASE(1k)	SASE	5.097	7.133	3.64

synthetic data have been passed to the feature extractor module. The MSE loss of the Head Pose Regressor module is calculated against the labelled head pose synthetic data and the classifier binary cross-entropy loss is measured against the binary labelled synthetic and real data (Biwi). The same second phase experiment is also conducted with the real dataset SASE. The trained model is then evaluated against the Biwi and SASE datasets.

Table 5 shows the comparative result with and without the domain adaptation for the two real-world datasets. The result shows that applying adversarial domain adaptation-based training improves the result by 1° across yaw, pitch and roll. Also, the predicted label and the ground truth label distribution is plotted in a scatter plot and shown in figure 11.

TABLE 5. Comparative result on Biwi and sase dataset with and without domain adaptation.

Strategy	MAE	Yaw	Pitch	Roll
Without domain adaptation on Biwi	6.34	5.86	6.51	6.63
With domain adaptation on Biwi	5.13	4.876	5.915	5.28
Without domain adaptation on SASE	6.633	6.523	7.769	5.61
With domain adaptation on SASE	6.04	5.135	7.28	5.32

VIII. DISCUSSION

The following section discusses the results presented in the previous section.

- In the first set of experiments, the model is trained with only the synthetic data and evaluated against Biwi. The result shows that the trained model performs close to the state-of-the-art. A similar result is found when the model is evaluated against the narrow band of yaw, pitch and roll as shown in table 2. Only for the high concatenated rotation angles, the model fails to sufficiently predict, and the errors are large. The first row of figure 11 shows the distribution of the ground truth labels and the predicted labels. From the distribution, it can be seen that the trained model performs poorly on either higher values of pitch and roll or higher values of yaw and roll.

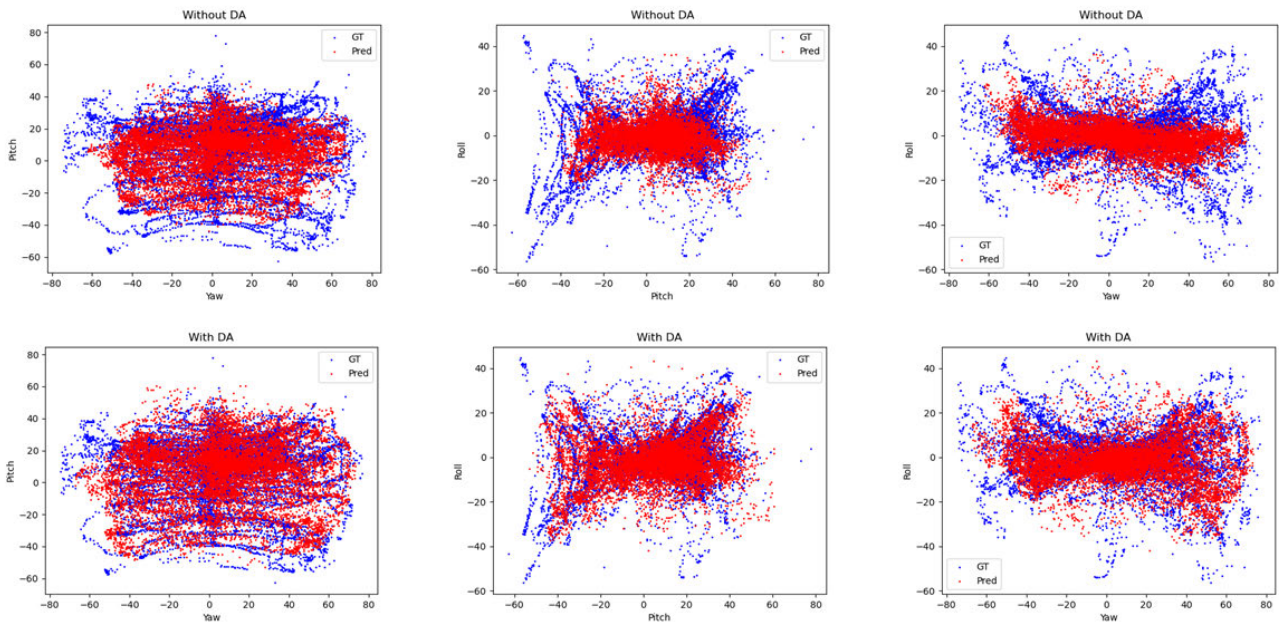


FIGURE 11. Distribution of ground truth and predicted labels in blue and red color respectively. The first row shows the result without domain adaptation and the second row shows with domain adaptation. The first column is Yaw versus Pitch, second column is Pitch versus Roll and the third column shows Yaw versus Roll label distribution.

TABLE 6. Experimental results on varying the background of the synthetic data and validating against Biwi.

Background	Test Set	MAE	Yaw	Pitch	Roll
Plain	Biwi	7.13	6.83	7.22	7.32
Textured	Biwi	6.84	6.39	7.35	6.77
Real	Biwi	7.08	6.63	7.71	6.9
Textured + Real	Biwi	6.34	5.86	6.51	6.63

- Though the model trained with the new synthetic data performs poorly in some extreme angles when it is compared with the previously available synthetic dataset Synhead, it performs better and produces good results overall as well in all the filtered angles as shown in table 3. A possible reason may be the lack of variation in the Synhead dataset, as it only contains 10 different subjects, whereas the synthetic data produced in this work has 100 subjects. Also, as the Synhead data is produced from a head scan, there are artefacts in some extreme angles compared to the proposed dataset as in this work the images are rendered from fully rigged full-body models. A few samples are shown in appendix B.
- In the data augmentation and data fusion-based transfer learning approach also the newly proposed synthetic data produces a better result than the previous work [32], where the model was trained on both real and synthetic data and tested on a set of both synthetic and real data. During the training, Wang et al. [32] have used around 200k of synthetic data and 12k of real data from the Biwi dataset, whereas using the synthetic data produced by this work during the initial training and then

fine-tuning the trained model with only 1k of Biwi data is able to beat the result of [32].

- In the final set of experiments where the adversarial domain adaptation is applied, the model performs better than the first phase where the network is trained only on synthetic data. Therefore, we conclude that the domain adaptation technique helps to learn the domain invariant features from both the synthetic and real domain. From figure 11 it can be found that after applying DA the trained model is able to predict the head pose in those extreme angles (high yaw and roll or high pitch and roll) as well where the model trained without the DA fails.
- Finally, as the data has been generated with three different backgrounds – plain, textured and real, it has been observed that training with the data augmenting with textured and real background images gives the best result among the three. The detailed results are shown in appendix A.

IX. CONCLUSION AND FUTURE WORK

In this article, a framework is presented to generate synthetic head pose data with their ground truth using a low-cost open-source toolchain, compared to previous works that generated synthetic datasets from expensive high-quality 3D scans. By generating the data with enough variations and covering real data distributions, we can achieve near state-of-the-art results training only with low-cost synthetic data. When compared with the previously available synthetic datasets, experimental results show that training a state-of-the-art HPE model with the data produced by this work gives better results in multiple scenarios. First, when the model is trained only



FIGURE 12. Samples from SynHead [7] dataset with artefacts because of large-concatenated rotation angles and samples from the dataset produced from this work with similar head rotations.

with synthetic data it gives a better result than the previous available dataset SynHead [7]. In the second scenario when the model is first trained on synthetic data and further fine-tuned with a very small amount of real data through transfer learning it produces a superior result than the previous work [32]. Further, it has been shown that applying the synthetic to real domain adaptation technique with adversarial training can reduce the gap between the synthetic and real domain and enables to learn the domain invariant features which further improve the result.

In future work, the proposed methodology can be used to bring these fully rigged models to various synthetic complex environments and build datasets for more specific tasks like in-cabin driver monitoring systems. As the head pose ground truth collected through this methodology is perfect without any error, cross-validation with the existing real head pose datasets can be performed by training the HPE model with various real dataset and validating against the synthetic data and vice-versa. The results can then be analysed to identify the errors in the ground truth of the real head pose datasets, particularly for large-concatenated head rotation angles. Additionally, as these full-body models are fully rigged and all the body parts can be accessed, more complex datasets can be created for human action sequences, facial gestures and dynamic head-pose sequences. Finally, the unsupervised domain adversarial learning is mostly used for classification tasks and not widely examined for continuous value prediction through regression, so the Domain Adaptation can further be examined for other regression tasks such as single view depth estimation and surface normal prediction while training on data from another domain (synthetic data).

APPENDIX A

Table 6 shows the comparative result of the FSA-Net trained on data generated by the methodology proposed in this work with three different backgrounds. The result shows combining the data with real and textured background produces the best result.

APPENDIX B

Figure 12 shows some of the examples from the SynHead [7] dataset with high values of pitch and yaw. As these are generated from single head scans and contain single mesh without any rigging there are some artefacts in those extreme angles.

In contrast in this work, a fully rigged full-body model is used, so there are no similar artefacts after rendering the models.

REFERENCES

- [1] E. N. A. Neto, R. M. Duarte, R. M. Barreto, J. P. Magalhães, C. C. M. Bastos, T. I. Ren, and G. D. C. Cavalcanti, "Enhanced real-time head pose estimation system for mobile device," *Integr. Comput.-Aided Eng.*, vol. 21, no. 3, pp. 281–293, Apr. 2014.
- [2] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool, "Random forests for real time 3D face analysis," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 437–458, Feb. 2013, doi: [10.1007/s11263-012-0549-0](https://doi.org/10.1007/s11263-012-0549-0).
- [3] G. Fanelli, T. Weise, J. Gall, and L. Van Gool, "Real time head pose estimation from consumer depth cameras," in *Proc. Joint Pattern Recognit. Symp.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 6835, 2011, pp. 101–110, doi: [10.1007/978-3-642-23123-0_11](https://doi.org/10.1007/978-3-642-23123-0_11).
- [4] G. P. Meyer, S. Gupta, I. Frosio, D. Reddy, and J. Kautz, "Robust model-based 3D head pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3649–3657.
- [5] M. Martin, F. Van De Camp, and R. Stiefelwagen, "Real time head model creation and head pose estimation on consumer depth cameras," in *Proc. 2nd Int. Conf. 3D Vis.*, Dec. 2014, pp. 641–648, doi: [10.1109/3DV.2014.54](https://doi.org/10.1109/3DV.2014.54).
- [6] G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara, "POSEidon: Face-from-depth for driver pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5494–5503, doi: [10.1109/CVPR.2017.583](https://doi.org/10.1109/CVPR.2017.583).
- [7] J. Gu, X. Yang, S. De Mello, and J. Kautz, "Dynamic facial analysis: From Bayesian filtering to recurrent neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1531–1540, doi: [10.1109/CVPR.2017.167](https://doi.org/10.1109/CVPR.2017.167).
- [8] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 2074–2083, doi: [10.1109/CVPRW.2018.00281](https://doi.org/10.1109/CVPRW.2018.00281).
- [9] T.-Y. Yang, Y.-T. Chen, Y.-Y. Lin, and Y.-Y. Chuang, "FSA-Net: Learning fine-grained structure aggregation for head pose estimation from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1087–1096, doi: [10.1109/CVPR.2019.00118](https://doi.org/10.1109/CVPR.2019.00118).
- [10] A. Berg, M. Oskarsson, and M. O'Connor, "Deep ordinal regression with label diversity," 2020, *arXiv:2006.15864*. [Online]. Available: <http://arxiv.org/abs/2006.15864>
- [11] N. Gourier, D. Hall, and J. L. Crowley, "Estimating face orientation from robust detection of salient facial structures," in *Proc. FG Net Workshop Vis. Observ. Deictic Gestures*, 2004, pp. 1–9.
- [12] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1785–1792, doi: [10.1109/CVPR.2011.5995702](https://doi.org/10.1109/CVPR.2011.5995702).
- [13] *Real-Time 3D Animation Software | iClone | Reallusion*. Accessed: Nov. 2, 2020. [Online]. Available: <https://www.reallusion.com/iclone/>
- [14] *Blender—Home of the Blender Project—Free and Open 3D Creation Software*. Accessed: Nov. 10, 2020. [Online]. Available: <https://www.blender.org/>
- [15] A. Gee and R. Cipolla, "Determining the gaze of faces in images," *Image Vis. Comput.*, vol. 12, no. 10, pp. 639–647, Dec. 1994, doi: [10.1016/0262-8856\(94\)90039-6](https://doi.org/10.1016/0262-8856(94)90039-6).
- [16] A. Nikolaidis and I. Pitas, "Facial feature extraction and pose determination," *Pattern Recognit.*, vol. 33, no. 11, pp. 1783–1791, Nov. 2000, doi: [10.1016/S0031-3203\(99\)00176-4](https://doi.org/10.1016/S0031-3203(99)00176-4).
- [17] A. Narayanan, R. M. Kaimal, and K. Bijlani, "Yaw estimation using cylindrical and ellipsoidal face models," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2308–2320, Oct. 2014, doi: [10.1109/TITS.2014.2313371](https://doi.org/10.1109/TITS.2014.2313371).
- [18] L. Fridman, J. Lee, B. Reimer, and T. Victor, "'Owl' and 'Lizard': Patterns of head pose and eye pose in driver gaze classification," *IET Comput. Vis.*, vol. 10, no. 4, pp. 308–313, Jun. 2016, doi: [10.1049/iet-cvi.2015.0296](https://doi.org/10.1049/iet-cvi.2015.0296).
- [19] H. Yuan, M. Li, J. Hou, and J. Xiao, "Single image-based head pose estimation with spherical parametrization and 3D morphing," *Pattern Recognit.*, vol. 103, Jul. 2020, Art. no. 107316, doi: [10.1016/j.patcog.2020.107316](https://doi.org/10.1016/j.patcog.2020.107316).

- [20] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 300–311, Jun. 2010, doi: [10.1109/ITITS.2010.2044241](https://doi.org/10.1109/ITITS.2010.2044241).
- [21] S. O. Ba and J. Odobez, "Multiperson visual focus of attention from head pose and meeting contextual cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 101–116, Jan. 2011, doi: [10.1109/TPAMI.2010.69](https://doi.org/10.1109/TPAMI.2010.69).
- [22] D. J. Tan, F. Tombari, and N. Navab, "Real-time accurate 3D head tracking and pose estimation with consumer RGB-D cameras," *Int. J. Comput. Vis.*, vol. 126, nos. 2–4, pp. 158–183, Apr. 2018, doi: [10.1007/s11263-017-0988-8](https://doi.org/10.1007/s11263-017-0988-8).
- [23] H. Liang, J. Hou, J. Yuan, and D. Thalmann, "Random forest with suppressed leaves for Hough voting," in *Proc. Asian Conf. Comput. Vis.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 10113, 2017, pp. 264–280, doi: [10.1007/978-3-319-54187-7_18](https://doi.org/10.1007/978-3-319-54187-7_18).
- [24] G. Riegler, M. R  ther, and H. Bischof, "Hough networks for head pose estimation and facial feature localization," in *Proc. Brit. Mach. Vis. Conf.*, 2014, p. 1, doi: [10.5244/c.28.66](https://doi.org/10.5244/c.28.66).
- [25] A. K. Rajagopal, R. Subramanian, E. Ricci, R. L. Vieriu, O. Lanz, R. R. Kalpathi, and N. Sebe, "Exploring transfer learning approaches for head pose classification from multi-view surveillance images," *Int. J. Comput. Vis.*, vol. 109, nos. 1–2, pp. 146–167, Aug. 2014, doi: [10.1007/s11263-013-0692-2](https://doi.org/10.1007/s11263-013-0692-2).
- [26] C. Papazov, T. K. Marks, and M. Jones, "Real-time 3D head pose and facial landmark estimation from depth images using triangular surface patch features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4722–4730.
- [27] B. Ahn, J. Park, and I. S. Kweon, "Real-time head orientation from a monocular camera using deep neural network," in *Proc. Asian Conf. Comput. Vis.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 9005, 2015, pp. 82–96, doi: [10.1007/978-3-319-16811-1_6](https://doi.org/10.1007/978-3-319-16811-1_6).
- [28] M. Patacchiola and A. Cangelosi, "Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods," *Pattern Recognit.*, vol. 71, pp. 132–143, Nov. 2017, doi: [10.1016/j.patcog.2017.06.009](https://doi.org/10.1016/j.patcog.2017.06.009).
- [29] F.-J. Chang, A. T. Tran, T. Hassner, I. Masi, R. Nevatia, and G. Medioni, "FacePoseNet: Making a case for landmark-free face alignment," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 1599–1608.
- [30] T.-Y. Yang, Y.-H. Huang, Y.-Y. Lin, P.-C. Hsiu, and Y.-Y. Chuang, "SSR-Net: A compact soft stagewise regression network for age estimation," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2018, p. 7, doi: [10.24963/ijcai.2018/150](https://doi.org/10.24963/ijcai.2018/150).
- [31] *3D Models | 3D Models From 3D Scans | 3Dscanstore*. Accessed: Nov. 4, 2020. [Online]. Available: <https://www.3dscanstore.com/>
- [32] Y. Wang, W. Liang, J. Shen, Y. Jia, and L.-F. Yu, "A deep coarse-to-fine network for head pose estimation from synthetic data," *Pattern Recognit.*, vol. 94, pp. 196–206, Oct. 2019, doi: [10.1016/j.patcog.2019.05.026](https://doi.org/10.1016/j.patcog.2019.05.026).
- [33] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, "Face alignment across large poses: A 3D solution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 146–155, doi: [10.1109/CVPR.2016.23](https://doi.org/10.1109/CVPR.2016.23).
- [34] M. Kostinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 2144–2151, doi: [10.1109/ICCVW.2011.6130513](https://doi.org/10.1109/ICCVW.2011.6130513).
- [35] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 1992, pp. 335–343.
- [36] A. Savran, N. Aly  z, H. Dibeklioglu, O.   elikutan, B. G  kberk, B. Sankur, and L. Akarun, "Bosphorus database for 3D face analysis," in *Proc. Eur. Workshop Biometrics Identity Manage.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 5372, 2008, pp. 47–56, doi: [10.1007/978-3-540-89991-4_6](https://doi.org/10.1007/978-3-540-89991-4_6).
- [37] I. L  si, S. Escarela, and G. Anbarjafari, "SASE: RGB-depth database for human head pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, 2016, pp. 325–336, doi: [10.1007/978-3-319-49409-8_26](https://doi.org/10.1007/978-3-319-49409-8_26).
- [38] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by back-propagation," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 1180–1189.
- [39] Z. Cao, L. Ma, M. Long, and J. Wang, "Partial adversarial domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, 2018, pp. 139–155, doi: [10.1007/978-3-030-01237-3_9](https://doi.org/10.1007/978-3-030-01237-3_9).
- [40] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8156–8164, doi: [10.1109/CVPR.2018.00851](https://doi.org/10.1109/CVPR.2018.00851).
- [41] Q. Chen, Y. Liu, Z. Wang, I. Wassell, and K. Chetty, "Re-weighted adversarial adaptation network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7976–7985, doi: [10.1109/CVPR.2018.00832](https://doi.org/10.1109/CVPR.2018.00832).
- [42] F. Kuhnke and J. Ostermann, "Deep head pose estimation using synthetic images and partial adversarial domain adaption for continuous label spaces," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10164–10173, doi: [10.1109/ICCV.2019.01026](https://doi.org/10.1109/ICCV.2019.01026).
- [43] *Character Creator—Fast Create Realistic and Stylized Characters*. Accessed: Nov. 2, 2020. [Online]. Available: <https://www.reallusion.com/character-creator/>
- [44] E. P. Lafortune and Y. D. Willems, "Bi-directional path tracing," in *Proc. SIGGRAPH*, 1993, pp. 1–8.
- [45] S. Basak, H. Javidnia, F. Khan, R. McDonnell, and M. Schukat, "Methodology for building synthetic datasets with virtual humans," in *Proc. 31st Irish Signals Syst. Conf. (ISSC)*, Jun. 2020, pp. 1–6, doi: [10.1109/ISSC49989.2020.9180188](https://doi.org/10.1109/ISSC49989.2020.9180188).
- [46] S. Abdelmounaime and H. Dong-Chen, "New brodatz-based image databases for grayscale color and multiband texture analysis," *ISRN Mach. Vis.*, vol. 2013, pp. 1–14, Feb. 2013, doi: [10.1155/2013/876386](https://doi.org/10.1155/2013/876386).
- [47] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1–9.
- [48] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.
- [49] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016, doi: [10.1109/LSP.2016.2603342](https://doi.org/10.1109/LSP.2016.2603342).
- [50] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem? (And a dataset of 230,000 3D facial landmarks)," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1021–1030, doi: [10.1109/ICCV.2017.116](https://doi.org/10.1109/ICCV.2017.116).



includes deep learning tasks related to computer vision.



He is currently an IEEE Fellow recognized for his contributions to digital camera technologies, notably in-camera red-eye correction and facial detection. He is a member of the IEEE Consumer Electronics Society for more than 25 years. He is also the Editor-in-Chief and the Founding Editor of *IEEE Consumer Electronics Magazine*.

SHUBHAJIT BASAK received the B.Tech. degree in electronics and communication engineering from the West Bengal University of Technology, India, in 2011, and the M.Sc. degree in computer science from the National University of Ireland Galway, Ireland, in 2018, where he is currently pursuing the Ph.D. degree in computer science. He has more than six years of industrial experience as a Software Development Professional. He is also with FotoNation/Xperi. His research interest

PETER CORCORAN (Fellow, IEEE) holds the Personal Chair in electronic engineering at the College of Science and Engineering, National University of Ireland Galway. He was the Co-Founder in several start-up companies, notably FotoNation, now the Imaging Division of Xperi Corporation. He has more than 600 technical publications and patents, more than 120 peer-reviewed journal articles, 150 international conference papers, and a co-inventor of more than 300 granted U.S. patents.



FAISAL KHAN received the B.S. degree in mathematics from the University of Malakand, Chakdara, Pakistan, in 2015, and the M.Phil. degree in mathematics from Hazara University, Mansehra, Pakistan, in 2017. He is currently pursuing the Ph.D. degree with the National University of Ireland Galway (NUIG). He is also with FotoNation/Xperi. His research interest includes machine learning using deep neural networks for tasks related to computer vision, including depth estimation and 3-D reconstruction.



RACHEL MCDONNELL is currently an Associate Professor of creative technologies with the School of Computer Science and Statistics, Trinity College Dublin. She combines research in cutting-edge computer graphics and investigating the perception of virtual characters to both deepen our understanding of how virtual humans are perceived, and directly provide new algorithms and guidelines for industry developers on where to focus their efforts. She has published more than 70 papers in the top conferences and journals in her field. She has served as an Associate Editor for *ACM Transactions on Applied Perception* and the *Journal of Eurographics*, the European Association for Computer Graphics.



MICHAEL SCHUKAT (Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science and medical informatics from the University of Hildesheim, Germany, in 1994 and 2000, respectively. He is currently a Lecturer and a Researcher with the School of Computer Science, National University of Ireland Galway, Galway. From 1994 to 2002, he has worked in various industry positions, where he specialized in deeply embedded real-time systems across diverse domains, such as industrial control, medical devices, automotive, and network storage. His research interests include AI and its application in computer vision, cybersecurity, health informatics, and energy management.

• • •

Learning Accurate Head Pose for Consumer Technology From 3D Synthetic Data

Shubhajit Basak
College of Engineering and Informatics
National University of Ireland,
Galway
Galway, Ireland
s.basak1@nuigalway.ie

Faisal Khan
College of Engineering and Informatics
National University of Ireland,
Galway
Galway, Ireland
f.khan4@nuigalway.ie

Rachel McDonnell
School of Computer Science and Statistics
Trinity College Dublin
Dublin, Ireland
ramcdonn@scss.tcd.ie

Michael Schukat
College of Engineering and Informatics
National University of Ireland,
Galway
Galway, Ireland
michael.schukat@nuigalway.ie

Abstract— Accurate 3D head pose estimation from a 2D image frame is an essential component of modern consumer technology (CT). It enables a better determination of user attentiveness and engagement and can support immersive audio and AR experiences. While deep learning methods have improved the accuracy of head pose estimation models, these depend on the accurate annotation of training data. The acquisition of real-world head pose data with a large variation of yaw, pitch and roll is a very challenging task. Available head-pose datasets often have limitations in terms of the number of data samples, image resolution, annotation accuracy and sample diversity (gender, race, age). In this work, a rendering pipeline is proposed to generate pixel-perfect synthetic 2D headshot images from high-quality 3D facial models with accurate pose angle annotations. A diverse range of variations in age, race, and gender are provided. The resulting dataset includes more than 300k pairs of RGB images with the corresponding head pose annotations. For every hundred 3D models there are multiple variations in pose, illumination and background. The dataset is evaluated by training a state-of-the-art head pose estimation model and testing against the popular evaluation dataset BIWI. The results show training with purely synthetic data produced by the proposed methodology can achieve close to state-of-the-art results on the head pose estimation task and is better generalized for age, gender and racial diversity than solutions trained on ‘real-world’ datasets.

Keywords— *Head Pose Estimation, Synthetic Face, Face Dataset*

I. INTRODUCTION

Head pose estimation (HPE) has great potential to provide an enabling technology for many next-generation consumer technologies (CT) including virtual reality (VR) and augmented reality (AR) based entertainment systems, human-computer interfaces (HCI) that employ human behaviour or attentiveness analysis, driver monitoring systems (DMS), and immersive audio systems. In human behaviour analysis, HPE is used for estimating human gaze and body posture to infer the feelings, desires etc. of a human subject. Facial authentication software can use HPE to improve performance and robustness. In DMS a real-time HPE is important to monitor the driver attention level, cognitive state and track eye-movements and gaze direction. For

AR/VR application HPE can be used to predict the accurate field of view (FOV) and is essential for foveated rendering in VR headsets.

Computer-vision based HPE transforms the captured 2D facial images into high-level directional data in three-dimensional space with three Euler angles: θ_x (Pitch), θ_y (Yaw) and θ_z (Roll). Normally the HPE tasks follow two different approaches: classification and regression. Regression approaches predict the head pose by fitting a regression model on the training data and estimating the yaw, pitch and roll in continuous angles, making these models comparatively complex. On the other hand, classification approaches mostly rely on classifying the head pose into a discrete bin. These methods are comparatively robust to large pose variations but with sparse solution space e.g. 10 degrees intervals for each bin.

Head pose estimation from a single image makes the problem more challenging. It requires learning the mapping between 3D and 2D spaces. Previous works use different modalities like depth information [1, 2, 3, 4], video sequences [6] or inertial measurement unit (IMU) [5]. An accurate depth map provides additional 3D cues that are missing in 2D images and requires expensive depth sensors. Most of this single image-based HPE methods leverage the use of Convolution Neural Network (CNN), a variant of a Deep Neural Network (DNN) to extract features from the 2D images and use those high-level features to model 3D head pose regressors. The recent state of the art models [7, 8, 9] shows combining the robustness of the classifier with the sensitivity of the regressor networks through a fine-to-coarse approach that makes these models more accurate.

Though these DNN based methods have given good results, a major drawback of these supervised models is their need for accurately labelled data. Particularly for HPE tasks, it will become more challenging to obtain annotated head pose data with variations of appearances like race, age, gender and other environmental factors like noise, illumination and occlusion. Also, obtaining real human data falls under different data protection and ethical guidelines like GDPR. Other modalities such as depth and IMU are prone to sensor noise. The head-pose

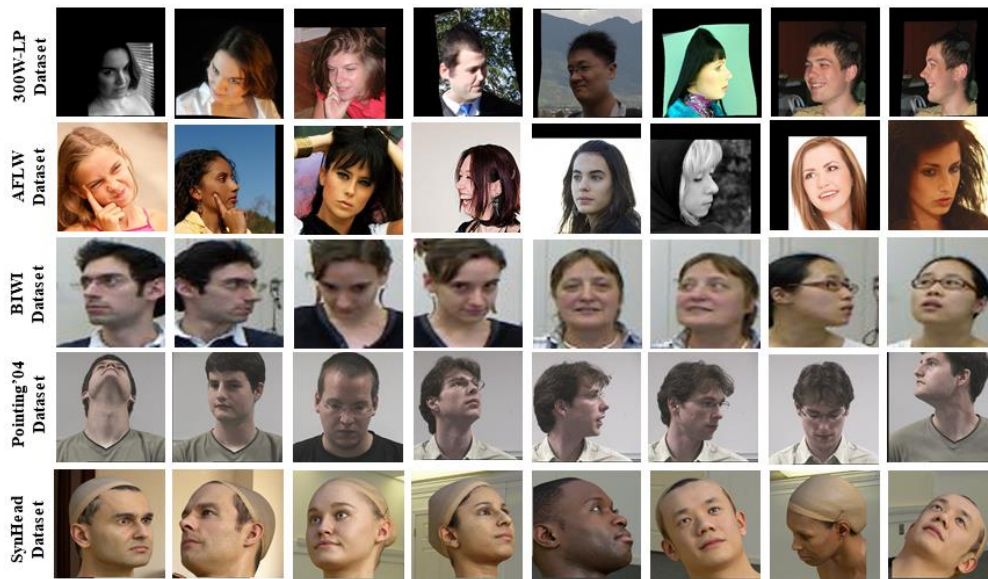


Figure 1. Sample Images from different datasets

datasets available captured from real subjects like BIWI Kinect Head Pose Dataset [1] and Pointing'04 [10] only consists around 15k and 4k images respectively. Among these two BIWI is most commonly used for benchmarking. But because of the limited size, both these datasets are not suitable to train DNN based HPE models. Generating synthetic facial images using Computer Graphics (CG) software provides a powerful tool for building large datasets of accurately labelled 2D facial image samples.

In this paper, we propose a methodology utilizing commercially available animation software and open-source CG tools to create photorealistic virtual human models and generate accurate RGB and corresponding ground truth Head Pose data. The data generated through this method is also been evaluated using the current state-of-the-art models. Training only on the synthetic dataset and testing on real dataset shows promising results except for some marginal areas of the data distribution.

II. RELATED WORKS

In this section, first deep learning-based HPE methods have been reviewed, before reviewing the currently available head pose datasets.

A. Head Pose Estimation using Deep Learning

Head Pose Estimation from visual information can be categorised into a few approaches. The first one is the facial geometric landmark-based method where these facial features have been used to fit appearance-based head models [12, 13] to calculate the accurate head pose. Different regression methods [14, 15] creates initial face models from the key points and incrementally align the created face with real ones by regressions. A comprehensive survey of these conventional methods can be found in [11]. As these landmark-based approaches require manual annotation of the landmarks in faces, it is often difficult to acquire such labels. In some cases, because of the low resolution of the images, accurately locating these landmarks is not possible.

Other approaches take advantage of different modalities as well. Fanelli et al.[1] fits a regression random forest model to predict the head pose from the depth information. Meyer et al. [3] fits 3D morphable models to the depth images and regress the head pose from that. Gu et al.[6] propose the facial landmark features tracking by Recurrent Neural Network (RNN) using a sequence of RGB images from facial video using temporal cues.

Finally, there is another set of approaches which focuses on deep learning-based HPE from a single monocular RGB image. In this paper, we have used this approach to validate our data. The initial work on this was proposed by Anh et al. [16] which uses CNN based models to regress the head pose information. Cangelosi and Patacchiola [17] examine adaptive gradient methods with different CNN architectures for HPE tasks. Chang et al.[18] predicted the head pose and facial key points jointly using the ResNet model. Ruiz et al. [9] used ResNet50 backbone architecture for feature extraction and combined loss stream of regression and binned pose classification. Yang et al. [8] propose FSA-Net, a lightweight structure for head pose feature regression, using the stage-wise regression model SSR-Net [19].

Few of the above-mentioned works use synthetic facial images with the ground truth head pose to train their models. Ruiz et al. and Yang et al. use a synthetically expanded dataset 300W-LP, which is created by augmenting real images. Gu et al.[6] introduced the synthetically created dataset SynHead, which has been rendered through a CG tool from a very high-quality 3D scan obtained from [20]. They use a transfer learning approach and train the network on synthetic data and fine-tune with real data. Wang et al. [21] also introduce a synthetically rendered head pose dataset from high-quality 3D scans and propose a fine to a coarse network to predict accurate head pose. Though the data is not publicly available. They train their model with approx. 260k synthetic images from their dataset and 15k real images from the BIWI dataset. Kuhnke et al.[22] propose an Adversarial Synthetic to Real Domain Adaptation technique and uses the SynHead to train the network. This is the only work

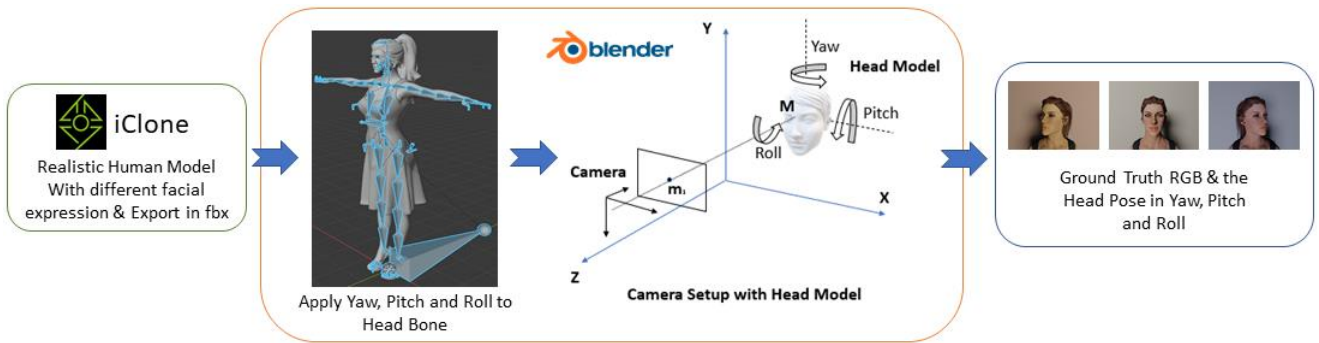


Figure 2. Overall Pipeline to produce the synthetic Head Pose Data

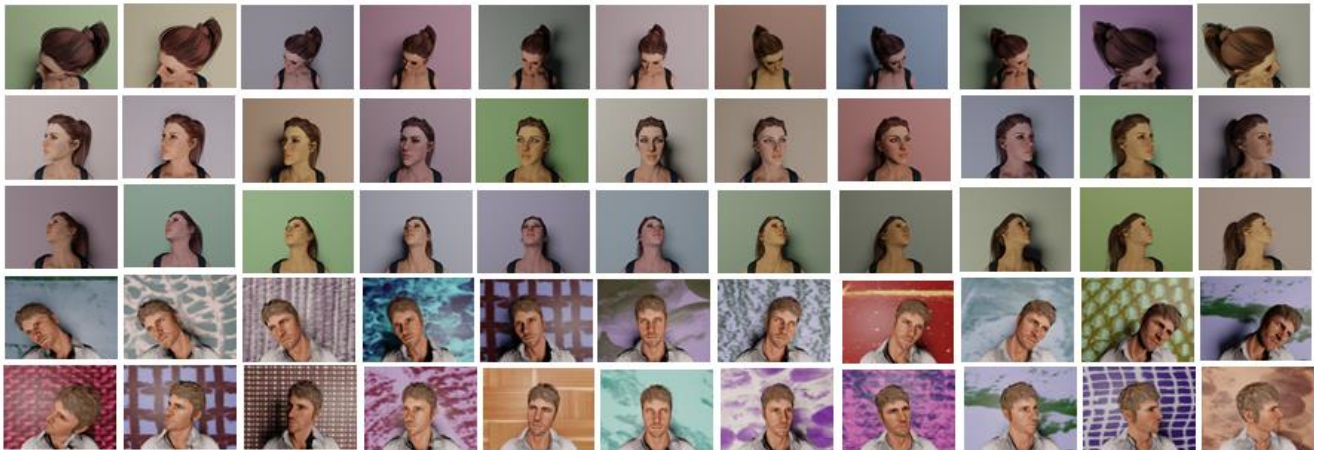


Figure 3. Samples from our dataset with plain and textured background and varying Yaw, Pitch, and Roll

which has trained on only synthetic data rendered from CG tool and tested on real data.

B. Head Pose Datasets

There are few datasets available which have been used for Monocular Image-based HPE tasks. Figure 1 shows the samples from these datasets.

300W-LP: 300W [23] uses multiple alignment real face databases with 68 facial key points including LFPW, AFW, IBUG, HELEN and XM2VTS. It uses 3D Dense Face Alignment (3DDFA) in which a dense 3D Face model is fitted to the images through a CNN which align faces in large poses up to 90 degrees. It contains around 61225 samples with large poses, which is further expanded to 122450 samples by flipping. The combined dataset is called 300W across Large Pose (300W-LP)

AFLW: AFLW [23] contains 21080 real faces in the wild with wide pose variations (yaw from -90 degree to +90 degree).

BIWI: Biwi Kinect Head Pose Dataset [1] contains approximately 15.7k images taken from 24 sequences of 20 subjects (12 men and 6 women, 4 people wearing glasses). Each image has a resolution of 640X480 pixels with the faces containing 90X110 pixel on average. The head pose ranges from $\pm 75^\circ$ yaw, $\pm 60^\circ$ pitch and $\pm 50^\circ$ roll.

Pointing'04: Pointing'04 [10] has been captured from 14 subjects containing 2.7k images. The head pose of the captured subjects is only represented by the two angles yaw and pitch and

both have fixed interval of 15 degrees. In our investigating we have found that during data acquisition the subjects have been asked to stare to different markers fixed in the room, resulting in an error in the captured labelled head rotation values for many samples. The pre-trained model of the current state-of-the-art HPE FSA-Net gives a Mean Absolute Error [MAE] of around 12 degrees while testing on this dataset.

SynHead: NVIDIA SynHead [6] contains 510960 frames of 70 head motion tracker rendered using 10 individual high-quality 3D scan head models from [20]. It contains head motion tracks of all 24 BIWI sequences. Though it was rendered with a different sequence of the rotation that was followed by BIWI.

Out of these datasets, because of their limitations of size, only the 300W-LP dataset is suitable for DNN training. Even though the SynHead Dataset has a large number of synthetic head pose frames, it only contains 10 individual subjects from high-quality 3D scans, which make it less diverse expensive to acquire. On the contrary our dataset has more than 300k frames from 100 individual models.

III. METHODOLOGY & DATASET DETAILS

In this section, we discuss the detailed methodology of creating the synthetic dataset which includes the RGB images and the corresponding ground truth head pose. Later we provide dataset details and analysis on the generated dataset.

A. 3D Model and Scene Setup

To generate the virtual human models, we have used the commercially available software iClone 7 and Character Creator [24]. The Character Creator comes with a “Realistic Human 100” package consisting of 100 human models with different age, race, gender, and ethnicity, thus reducing the bias of the dataset. Additionally, the facial morphs and expressions are also adjusted to provide more variations. All these models are exported from iClone in FBX formats with Physically Based Rendering (PBR) textures to add realism to them. These fully rigged models in FBX formats are then imported in open-source 3D creation software Blender [25]. The FBX models contain the fully rigged armature with the mesh which can be used to add motions to the head. To vary the scene light, we have added different illuminations available in Blender, which includes point, area, sun, and spotlight. To render the actual image, a camera model has been added to the scene in perspective mode. We have chosen the Blender cycle rendering engine which provides the ray path tracing for realistic rendering. The detailed methodology can be found in [26]. To add variations to the background we have combined plain, textured, and real images. For the textured background, we have used the Brodatz-based colour images provided by [27]. For the real background, we have used the images provided by the SynHead [6] dataset in the background folder.

B. Applying Head Pose & Collect Ground Truth

As these models are fully rigged, the shoulder bone has been selected to provide the rotation to the head mesh. An empty object has been added to the centre of the two eyeballs which we have chosen as the centre of the head. The translation and the rotation of the main head bone have been copied to the empty object which constraint the empty to follow the head. The rotation has been applied to the head bone in the sequence of PRY (pitch, roll and yaw) and all the frames have been saved. We have varied the Yaw, Pitch and Roll in the range of $\pm 80^\circ$, $\pm 70^\circ$ and $\pm 55^\circ$, respectively in an interval of 3° . Additionally, we have also applied the Euler angles provided by the 24 Biwi sequences and recorded those frames as well. But as these models are rigged with the head mesh, for each frame the alignment is not exactly the same as Biwi. The mean average error with Biwi for these sequences is approx. 1° in Euler scale.

To render the ground truth the camera near and far clip parameters are set to 0.001 and 5.0 meters, respectively. The camera sensor size and field of view (FOV) are set at 60° and 36 millimetres. To get the final render the RGB render pass has been used in the Blender compositor setup. While rendering the frames saved previously the empty object’s current translation in Blender 3D world coordinate and rotation in Euler has been captured through an automated python script.

The rendering of ground truth is carried out in an Intel Core i7-6800 3.4 GHz 6 core CPU machine with 32 GB of RAM and two NVIDIA TITAN X Pascal Graphical Processing Unit (GPU) with 32 GB of dedicated graphics memory. The ground truth head pose RGB images are rendered with a resolution of 640×480 pixels in jpeg format. Each frame took 16.3 seconds in an average to render using Blender Ray path Tracing Cycle Rendering Engine.

The overall pipeline for generating the synthetic head pose has been shown in figure 2.

C. Dataset Details

Following the above-discussed methodology, we have generated the ground truth RGB images and their corresponding headpose (Pitch, Roll and Yaw) in Euler angle for 44 female and 56 male models. Each subject has approx. 3.5k samples which make the total dataset size to around 3,500k. A sample of images from the generated data with varying Yaw, Pitch and Roll has been shown in figure 3. While training a deep neural network, the generalization of the model highly depends on the data distribution of the dataset. So, to check the label distribution we randomly select a few identities from our dataset and compare them with the Biwi dataset. Figure 4 shows the two distributions which show our dataset is more uniform across the value of yaw, pitch, and roll, whereas the distribution of Biwi shows it is mainly concentrated on the angles near the centre.

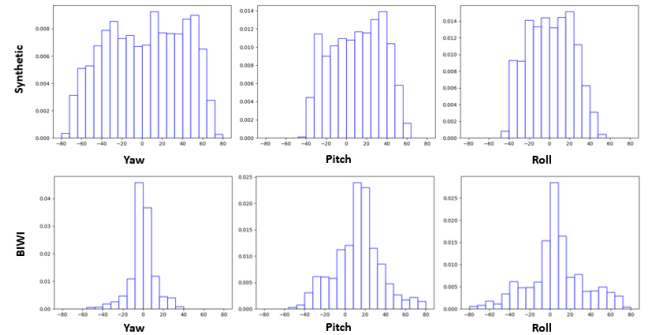


Figure 4. The first row shows the data distribution of Yaw, Pitch and Roll in our synthetic dataset and the second row shows the same distribution from Biwi Test dataset

IV. EVALUATION

In this section, we will first discuss one of the current state-of-the-art HPE models that we have used to evaluate our data. Later we will show the results of that model on our dataset.

A. Model Details

To evaluate our data, we have selected the recent state-of-the-art models FSA-Net [8], which has been trained on 300W-LP and Biwi in its original work and has been validated against Biwi. The FSA-Net model is based on feature aggregation and a soft stagewise regression based on previous work on SSR-Net [24] which employs a coarse-to-fine strategy for classification following the stage-wise regression. The soft stagewise regression (SSR) function accepts N set of stage parameters $\{\vec{p}^{(n)}, \vec{\eta}^{(n)}, A_n\}$.

1) *Feature Aggregation Module*: FSA-Net employs a spatial grouping of features and feeds it to the aggregation module. The feature map U_n for the nth stage is a spatial grid containing the k dimensional feature representation of a particular spatial location. Then it computes an attention map A_n through a scoring function, which helps to get the pixel-level feature. The original work deals with three different scoring options (1) Uniform, (2) 1×1 convolution and (3) Variance. We have used the third option, in which the features

TABLE I. EXPERIMENTAL RESULTS

Experiment	Model	Training Set	Test Set	MAE	Yaw	Pitch	Roll	
Intra Domain	Gu [6]	VGG16 [29]	Biwi	Biwi	3.66	3.91	4.03	3.03
	Ruiz [9]	ResNet50	Biwi	Biwi	3.23	3.29	3.39	3.00
	Yang [8]	FSA-Net Fusion	Biwi	Biwi	3.6	2.89	4.29	3.6
Inter Domain (300W-LP as Training Set)	Ruiz [9]	ResNet50	300W-LP	Biwi	4.90	4.81	6.61	3.27
	Yang [8]	FSA-Net Fusion	300W-LP	Biwi	4.00	4.27	4.96	2.76
Transfer Learning + Data Fusion	Wang [21]	GoogleNet [30]	Synthetic + Biwi	Biwi	4.96	4.76	5.48	4.29
Inter Domain Train only on our Synthetic Data	Ours	FSA-Net Capsule	Our Syn Data	Biwi	6.10	5.1	6.64	6.56
	Ours	FSA-Net Capsule	Our Syn Data	Biwi Yaw (+60°, -60°) Pitch (+60°, -60°) Roll (+10, -10°)	4.88	4.375	5.59	4.67

are selected through Variance, which is differentiable but not learnable. After getting the feature map U_n and attention map A_n , a set of representative features \tilde{U}_n has been extracted through $\tilde{U}_n = S_n U_n$. S_n is a linear dimensionality reduction transformation which has been learned from the attention map A_n . This representative features \tilde{U}_n is then sent to the existing feature aggregation method capsule [31] to get the representative features V .

2) *SSR-Net Module*: The SSR-Net employs a coarse-to-fine architecture for classification following the soft stage wise regression. The classification sets to divide the task into several bins of head pose (yaw, pitch and roll). A shift vector $\vec{\eta}^{(n)}$ predict the center of each bin and the scale factor Δ_n defines the width of the bin. The SSR soft stagewise regression function accepts N set of stage parameters $\{\vec{p}^{(n)}, \vec{\eta}^{(n)}, \Delta_n\}$ where $\vec{p}^{(n)}$ is the probability distribution of the n th stage. These stage parameters are obtained from the final set of feature vector V of the feature aggregation module. The final regressor output of the head pose then thus obtained by

$$\tilde{y} = \sum_{n=1}^N \vec{p}^{(n)} \cdot \vec{\mu}^{(n)}$$

a) where $\vec{\mu}^{(n)}$ is a vector for representative values of head pose group and obtained from $\vec{\eta}^{(n)}$ and Δ_n .

3) *Loss function*: The ultimate goal of the HPE task is to find a representative function $F(x)$ which predicts the head pose \tilde{y} for an input image x . To find F we have used the most common loss function found in HPE literature, the mean absolute error (MAE) between the ground truth and predicted head poses –

$$L(y, \tilde{y}) = \frac{1}{M} \sum_{m=1}^M \|\tilde{y}_m - y_m\|$$

where $\tilde{y}_m = F(x_m)$ is the predicted pose for the image x_m and y_m is the corresponding ground truth.

B. Experimental Details

We have used Pytorch to implement the FSA-Net module. As the main objective is to evaluate the data generated by our

method to check if the data is close enough to the real-world data, we trained the model only with our synthetic data and tested on the two different real datasets Biwi. We have not used any further data augmentation or transfer learning approach during our training. The training set consists of 200k labelled synthetic images. We trained the network for 90 epochs with the Adam optimizer. The initial learning rate has been set to 0.0001, later the learning rate has been reduced gradually after 30 epochs by 0.1. The experiments have been performed in an Intel I7 CPU and an Nvidia TitanX GPU.

C. Results & Discussion

During the evaluation, after training the FSA-Net model with our synthetic data, we have tested the trained model against BIWI dataset, which we think are closest to our data in terms of appearance. We have used the popular face recogniser MTCNN [28] to exclude some of the extreme angles where the face is out of the frame and loosely cropped the facial region to create the test dataset.

Table I shows the experimental result with the current state-of-the-art models. We have divided the results into two category intra-domains where both the training and testing data are real and from the same domain. In the case of inter-domain, the models are trained with synthetic or synthetic like (300W-LP) or fusion of Real and Synthetic data. We have found the network trained only on our synthetic data gives state-of-the-art result for a low roll. But when there is a mix of high negative pitch and high roll the model got confused and give an ambiguous result. We believe this is mostly because of the hair particle textures for the synthetic data as the face is not visible properly in these frames. For high roll with little variation in yaw and pitch also it gives MAE of approx. 2°.

V. CONCLUSION

In this paper, we have presented a framework to generate synthetic head pose data with their ground truth using the available cheap and open-source toolchain. Previous works have used synthetic dataset which has been generated from high-quality 3D scans thus making them expensive. Also, either they have used transfer learning or data fusion approach to train their model or domain adaptation techniques to reduce the gap

between synthetic and real domain. We have also shown that generating the data with enough variations and covering the real data distribution we can achieve near state-of-the-art result just by training with low-cost synthetic data. Though our model does not perform well on the boundary value of roll and pitch we believe it can be improved further on applying proper domain adaptation techniques.

ACKNOWLEDGMENT

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (D-REAL) under Grant No. 18/CRT/6224. The author also acknowledges Professor Peter Corcoran for providing valuable input throughout this work.

REFERENCES

- [1] Fanelli, G., Dantone, M., Gall, J., Fossati, A., & Van Gool, L. (2013). Random forests for real time 3d face analysis. *International journal of computer vision*, 101(3), 437-458.
- [2] Fanelli, G., Weise, T., Gall, J., & Van Gool, L. (2011, August). Real time head pose estimation from consumer depth cameras. In *Joint pattern recognition symposium* (pp. 101-110). Springer, Berlin, Heidelberg.
- [3] Meyer, G. P., Gupta, S., Frosio, I., Reddy, D., & Kautz, J. (2015). Robust model-based 3d head pose estimation. In *Proceedings of the IEEE international conference on computer vision* (pp. 3649-3657).
- [4] Martin, M., Van De Camp, F., & Stiefelhagen, R. (2014, December). Real time head model creation and head pose estimation on consumer depth cameras. In *2014 2nd International Conference on 3D Vision (Vol. 1, pp. 641-648)*. IEEE.
- [5] Borghi, G., Venturelli, M., Vezzani, R., & Cucchiara, R. (2017). Poseidon: Face-from-depth for driver pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4661-4670).
- [6] Gu, J., Yang, X., De Mello, S., & Kautz, J. (2017). Dynamic facial analysis: From bayesian filtering to recurrent neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1548-1557).
- [7] Berg, A., Oskarsson, M., & O'Connor, M. (2020). Deep Ordinal Regression with Label Diversity. *arXiv preprint arXiv:2006.15864*.
- [8] Yang, T. Y., Chen, Y. T., Lin, Y. Y., & Chuang, Y. Y. (2019). Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1087-1096).
- [9] Ruiz, N., Chong, E., & Rehg, J. M. (2018). Fine-grained head pose estimation without keypoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 2074-2083).
- [10] Gourier, N., Hall, D., & Crowley, J. L. (2004, August). Estimating face orientation from robust detection of salient facial structures. In *FG Net workshop on visual observation of deictic gestures (Vol. 6, p. 7)*. FGnet (IST-2000-26434) Cambridge, UK.
- [11] Murphy-Chutorian, E., & Trivedi, M. M. (2008). Head pose estimation in computer vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 31(4), 607-626.
- [12] Matthews, I., & Baker, S. (2004). Active appearance models revisited. *International journal of computer vision*, 60(2), 135-164.
- [13] Liang, L., Xiao, R., Wen, F., & Sun, J. (2008, October). Face alignment via component-based discriminative search. In *European conference on computer vision* (pp. 72-85). Springer, Berlin, Heidelberg.
- [14] Cao, X., Wei, Y., Wen, F., & Sun, J. (2014). Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2), 177-190.
- [15] Xiong, X., & De la Torre, F. (2015). Global supervised descent method. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2664-2673).
- [16] Ahn, B., Park, J., & Kweon, I. S. (2014, November). Real-time head orientation from a monocular camera using deep neural network. In *Asian conference on computer vision* (pp. 82-96). Springer, Cham.
- [17] Patacchiola, M., & Cangelosi, A. (2017). Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition*, 71, 132-143.
- [18] Chang, F. J., Tuan Tran, A., Hassner, T., Masi, I., Nevatia, R., & Medioni, G. (2017). Faceposenet: Making a case for landmark-free face alignment. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 1599-1608).
- [19] Yang, T. Y., Huang, Y. H., Lin, Y. Y., Hsiu, P. C., & Chuang, Y. Y. (2018, July). SSR-Net: A Compact Soft Stagewise Regression Network for Age Estimation. In *IJCAI (Vol. 5, No. 6, p. 7)*.
- [20] 3dscanstore.com. 2020. 3D Models | 3D Models From 3D Scans | 3Dscanstore.Com. [online] Available at: <https://www.3dscanstore.com> Accessed 21 August 2020.
- [21] Wang, Y., Liang, W., Shen, J., Jia, Y., & Yu, L. F. (2019). A deep coarse-to-fine network for head pose estimation from synthetic data. *Pattern Recognition*, 94, 196-206.
- [22] Kuhnke, F., & Ostermann, J. (2019). Deep head pose estimation using synthetic images and partial adversarial domain adaption for continuous label spaces. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 10164-10173).
- [23] Zhu, X., Lei, Z., Liu, X., Shi, H., & Li, S. Z. (2016). Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 146-155).
- [24] 3D Animation Software: iClone: Reallusion. (n.d.). Retrieved August 27, 2020, from <https://www.reallusion.com/iclone>.
- [25] Foundation, B. (n.d.). Home of the Blender project - Free and Open 3D Creation Software. Retrieved August 27, 2020, from <https://www.blender.org>.
- [26] Basak, S., Javidnia, H., Khan, F., McDonnell, R., & Schukat, M. (2020, June). Methodology for Building Synthetic Datasets with Virtual Humans. In *2020 31st Irish Signals and Systems Conference (ISSC)* (pp. 1-6). IEEE.
- [27] Abdelmounaime, S., & Dong-Chen, H. (2013). New Brodatz-based image databases for grayscale color and multiband texture analysis. *ISRN Machine Vision*, 2013.
- [28] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10), 1499-1503.
- [29] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [30] Szegedy, Christian, et al. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
- [31] Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems* (pp. 3856-3866).

High-Accuracy Facial Depth Models derived from 3D Synthetic Data

Faisal Khan
College of Engineering and Informatics
National University Ireland Galway
Galway, Ireland
f.khan4@nuigalway.ie

Shubhajit Basak
College of Engineering and Informatics
National University Ireland Galway
Galway, Ireland
s.basak1@nuigalway.ie

Hossein Javidnia
ADAPT Center, O'Reilly Institute
Trinity College Dublin Ireland
25 Westland Row, Dublin, 2, Ireland
hossein.javidnia@adaptcentre.ie

Michael Schukat
College of Engineering and Informatics
National University Ireland Galway
Galway, Ireland
micheal.schukat@nuigalway.ie

Peter Corcoran
College of Engineering and Informatics
National University Ireland Galway
Galway, Ireland
peter.corcoran@nuigalway.ie

Abstract—In this paper, we explore how synthetically generated 3D face models can be used to construct a high-accuracy ground truth for depth. This allows us to train the Convolutional Neural Networks (CNN) to solve facial depth estimation problems. These models provide sophisticated controls over image variations including pose, illumination, facial expressions and camera position. 2D training samples can be rendered from these models, typically in RGB format, together with depth information. Using synthetic facial animations, a dynamic facial expression or facial action data can be rendered for a sequence of image frames together with ground truth depth and additional metadata such as head pose, light direction, etc. The synthetic data is used to train a CNN-based facial depth estimation system which is validated on both synthetic and real images. Potential fields of application include 3D reconstruction, driver monitoring systems, robotic vision systems, and advanced scene understanding.

Keywords—3D Facial models, Facial depth, Face attributes, Facial image dataset

I. INTRODUCTION

Estimating human shape, pose, motion and depth from images are fundamental challenges for many multimedia applications and provide information that can be leveraged to enhance quality and immersion in advanced consumer use cases. Examples include scene analysis & understanding, human behaviour analysis, driver monitoring for semi-autonomous driving, augmented reality systems and facial expression analysis and facial authentication. Today, state-of-art systems for these use cases will rely on highly optimized convolutional neural networks designed to run on low-power embedded hardware. Such solutions require large, high-quality training datasets.

Facial images, in particular, are at the core of many consumer multimedia systems. They exhibit rich variations in pose, hairstyle, expression, structure and their 2D appearance is affected by external factors such as lighting and camera location. Many face variations can be synthesized using existing advanced 3D tools such as iClone [1] and Blender [2]. Using these tools, it is feasible to generate a large number of synthetic images required for training Convolutional Neural Network (CNN) models. Rendering synthetic facial images would be highly useful for numerous tasks as it can provide enough realism to create various ground truth in terms of occlusions, depth, motion, body-part segmentation, camera and light direction.

The current generation of deep learning models requires the datasets to contain various information and accurate data for the training and evaluation process. The existing human facial datasets do not have the accurate depth information that defines the actual position of each facial element. The depth information in these datasets requires the manual description of the scene, which is an error-prone and time-consuming task especially dealing with video [3]. In such type of facial dataset, they are not sufficiently large and varied enough to learn the CNN models, as a consequence, they come with a low performance which restricts real-world applications [4-5].

Recently deep learning-based methodologies have significantly improved the performances of face recognition systems, Human-Computer Interaction (HCI), understanding of 3D scenes for autonomous driving and robotics. An accurate determination of depth within the 3D scene is an important element of these computer vision systems. New emerging applications such as 3D reconstruction, Driver Monitoring Systems (DMS), robotic vision systems for personal robots and advanced HCI modalities require further improvements in short-range depth analysis to better understand and engage with humans.

In this work, we present a method for generating advanced facial models with synthetic data. A method is proposed to generate facial depth information using 3D virtual human and iClone [1] character modelling software. The proposed method can be scaled to produce any number of synthetic facial data by controlling the face animations, scene and camera position.

The main contribution of this research is focused on facial image rendering with the corresponding ground truth depth information. Using the synthetically generated data, we can train CNNs to address the facial depth estimation problem. This approach can enrich the real-world facial datasets required for portrait depth estimation problem.

The rest of the paper is structured as follows: Section II discusses related work and Section III presents the facial models. The application of synthetic facial depth (evaluation) is studied in Section IV. Conclusion and further cautions are discussed in Section V.

II. RELATED WORK

Facial depth estimation is considered as one of the challenging issues in computer vision, human-computer

interaction and virtual reality. It is used in a wide range of applications which includes controlling 3D avatars, human object detection and human-robot interactions [6-11].

Synthetic human facial data is used frequently to augment real data for pose invariant face recognition. By using the 3D morphable model and Basel face model [13, 14], a pipeline is proposed to create synthetic faces [15]. A synthetic dataset for person identification is studied in [16, 17]. The authors used Blender [2] rendering engine to create different realistic illumination conditions including indoor and outdoor scenes and introduce a novel domain adaptation method that uses the synthetic data. In [13], FaceGen Modeller is used for generating facial ground truth using morphable models. In [19], a large-scale synthetic dataset called (SURREAL) is introduced where the images are rendered from 3D sequences of MoCap data. In [18], synthetic bodies are obtained by utilizing the SMPL body model [18]. This dataset contains more than 6 million frames with ground truth depth, pose and segmentation masks [19].

Very limited work is done on synthetic facial models to explore the field with the available 3D tools and other commercially available software. In this paper, we proposed a method that generates synthetic facial models with many variations in expressions. By controlling the facial animations, camera positions, light positions, body poses, scene illuminations and other scene parameters, the method can be scaled to generate any number of labelled data samples.

III. FACIAL DEPTH GENERATOR MODEL

Virtual human models are created using the “*Realistic Human 100*” models in iClone [1] software based on the following steps:

A. The iClone Character Creation Process

iClone character creator [1] is used to create the initial characters of the virtual human faces. The iClone character creator generates humanoid characters and offers a useful 3D rigging option. The facial animation-ready models can be customized with sculpting and morphs. The template of the “*Realistic Human 100*” models is applied to the base body in the character creator as shown in Fig. 1.



Fig. 1. A sample from the iClone Character creator.

B. Adding Facial Expressions to Character Models

The virtual human face models are imported from Character creator to iClone [1]. Further, different expressions are added to the face models to introduce variations such as neutral, angry, happy, sad and scared. Fig. 2, show an example of these expressions.



Fig. 2. A sample rendered images of iClone with different expressions (neutral, angry, happy, sad and scared).

C. Exporting Character Animations to Blender

The created virtual human face models are exported from iClone [1] to Blender [2] in FBX format as it provides appropriate rigging. FBX is a popular 3D file format for exchanging the 3D information as used by many 3D tools including Blender [2]. A sample of an iClone facial model with base body loaded in Blender [2] is shown in Fig. 3.



Fig. 3. iClone facial model with base body loaded in Blender.

D. Rendering 2D Image Data with Ground Truth Depth

In this work, the following steps are taken to obtain the final output. The cameras and lights are placed in a fixed position and the corresponding distance of the models are changed in the range of 700-1000 mm. The focal length and sensor size are set to 60mm and 36mm respectively. The facial models are rotated in the virtual scenes. Fig. 4 shows a sample

of the camera and light position with respect to the facial models in Blender [2].

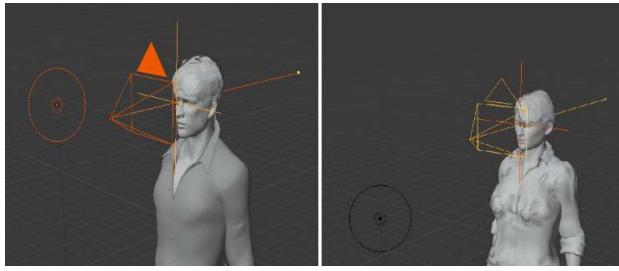


Fig. 4. A sample of the camera and light position with respect to the 3D character.

To generate RGB and depth images of faces in an extensive range of positions, the near and far clip of the camera is set to 0.01 and 5 meters. The facial models are rendered with 480×640 resolution and on a static background image. Fig. 5 shows a few rendered images while the camera position is changed with respect to the facial models.



Fig. 5. A facial model with corresponding ground truth depth of a head model from different views.

Fig. 6 illustrates facial models with the corresponding ground truth depth while the camera is positioned at different distances.



Fig. 6. A facial model with ground truth depth captured at the different camera position.

Render passes are set up in Blender [2] to generate the synthetic facial RGB and the corresponding ground truth depth images. To reduce the noise produced during the rendering process, the branched path tracing method is employed. Fig. 7 presents an overview of the noise controlling method in Blender [2].

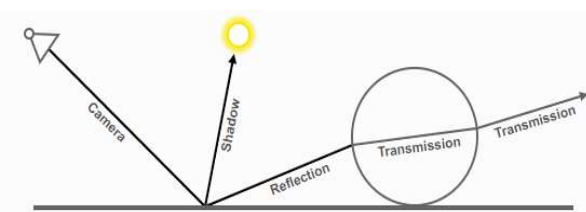


Fig. 7. An overview of the noise control system in Blender.

Afterwards, the images are rendered using Cycles engine and in the perspective view to obtain the RGB images with corresponding facial depth. Fig. 8 demonstrates the workflow of the facial depth generation process, camera and light setting.



Fig. 8. Rendering configuration in Blender. The left row shows the body shape, light and camera setting; the middle row shows the facial RGB and the last row illustrates the corresponding facial depth image.

Fig. 9 shows a few numbers of synthetic male and female models with the corresponding ground truth depth.



Fig. 9. A sample of the synthetic facial images with different expressions and their corresponding depth maps.

IV. EVALUATION

In this section, we deliver details about the evaluation of the two-state of the art CNNs on facial depth estimation. The pre-trained monocular depth estimation models DepthDense [19] and MiDas [20] are tested on the rendered synthetic data. Fig. 10, presents a few random synthetic RGB images and the corresponding depth images predicted using DepthDense [19]. Similarly, Fig. 11, shows the synthetic RGB images, predicted depth using MiDas [20] and ground truth images.



Fig. 10. Sample synthetic RGB images predicted depth maps by DepthDense [19] and corresponding ground truth.

TABLE I. RESULTS OF THE DEPTHDENSE, MIDAS MODELS [19, 20] AND SIMPLE CNN MODEL.

No.	Method	Abs Rel	Sq Rel	RMSE	RMSElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
1.	DenseDepth[19]	0.8765	0.7783	1.8783	0.2260	0.2723	0.5093	0.6912
2.	MiDas[20]	0.8876	0.9765	1.9876	0.3323	0.3211	0.5432	0.7635
3.	Simple CNN (full image)	0.0412	0.0123	0.0618	0.0177	0.9862	0.9971	0.9989
4.	Simple CNN (only face)	0.0370	0.0092	0.0196	0.0166	0.9961	0.9990	0.9979

^a Evaluation results of the pre-trained models [19, 20] on the synthetic data.

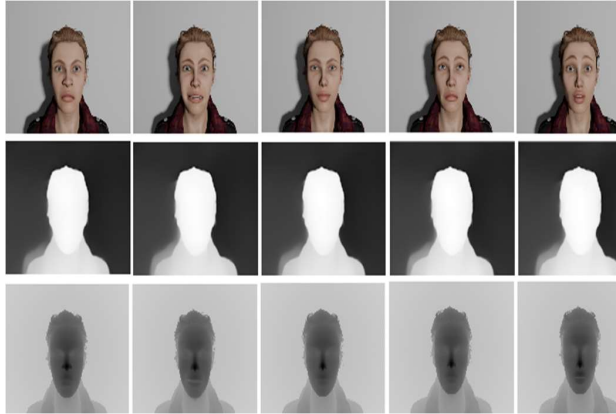


Fig. 11. Sample synthetic RGB images, predicted depth maps by MiDas [20] and corresponding ground truth.

The most common quantitative matrices for evaluating the performance of the pre-trained models including Absolute Relative difference (AbsRel), Root Mean Square Error (RMSE), log Root Mean Square Error (RMSE(log)) and Square Relative error (SqRel) are employed for evaluation purposes. Table 1, demonstrates the evaluation results of the DepthDense and MiDas models [19, 20].

To further evaluate the validity of the synthetic data generated in this paper, we re-trained a few recent CNN-based depth estimation networks [21, 22] on the generated facial data and later fine-tuned the models on real datasets.

A simple autoencoder with skip connection based on U-Net architecture has been trained using the data generated with a plain background as shown in Fig 12.

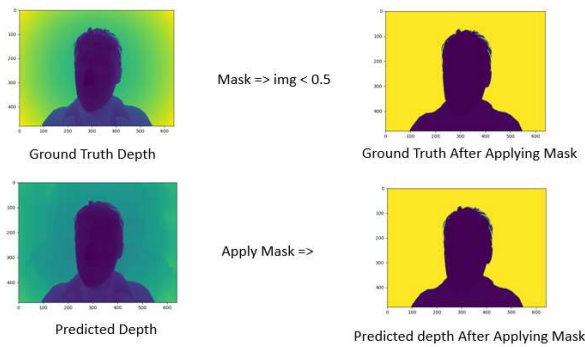


Fig. 12. Ground Truth Depth and Predicted Depth before and applying the mask.

Using the data generated with a plain background as shown in Fig 12, as a monocular depth estimation use case. There are around 40k training and 15k test images and their corresponding ground truth depth. The network has been initialized with random weight and trained with mean square error loss and Adam Optimiser. Further to evaluate the results only on a facial section of the image the depth has been masked within a range of 50 cm from the camera centre and the masked depth has been evaluated with the ground truth depth. Both the results have been shown in Table 1.

Furthermore, we will create additional variations and augmentations in the synthetic facial depth data to grow the final training dataset. It is expected that this will further increase the accuracy of these deep learning-based CNN networks when tested on real data.

V. CONCLUSION AND FUTURE RESEARCH

In this research paper, we proposed an advanced synthetic facial data generation pipeline. The facial images are generated from 3D virtual human models by rendering different variations of face poses, head poses and lighting conditions. Blender [2] rendering engine is used to generate the output as it allows changing different parameters such as lights position, camera parameters and keyframe values.

The proposed framework has the potential to generate a great number of synthetic facial images. The synthetic 3D models can be used in different 3D environments if scaled properly. This will allow simulating real-world scenarios by controlling the camera position, intrinsic parameters and lighting conditions.

The generated dataset can be used for training and validation of deep learning methods with the focus on natural face modelling, portrait 3D reconstruction and beautification.

In our future work, we will explore the potentials of the deep learning methods on direct facial 3D reconstruction using the synthetically generated data.

REFERENCES

- [1] 3D Animation Software: iClone: Reallusion. (n.d.). Retrieved from <https://www.reallusion.com/iclone/>.
- [2] Foundation, B. (n.d.). Home of the Blender project - Free and Open 3D Creation Software. Retrieved from <https://www.blender.org/>.
- [3] T. List, J. Bins, J. Vazquez, and R. B. Fisher. "Performance evaluating the evaluator". In ICCCN '05: Proceedings of the 14th International Conference on Computer Communications and Networks, pages 129–136, Washington, DC, USA, 2005. IEEE Computer Society.

- [4] S. R. Musse, R. Rodrigues, M. Paravisi, J. C. S. Jacques. Junior, and C. R. Jung. "Using synthetic ground truth data to evaluate computer vision techniques". In IEEE Workshop on Performance Evaluation of Tracking Systems (in conjunction with ICCV 07), pages 25–32, 2007.
- [5] G. R. Taylor, A. J. Chosak, and P. C. Brewer. "Using virtual worlds to design and evaluate surveillance systems". In Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, pages 1–8, 2007.
- [6] S.S. Mukherjee, N.M. Robertson, "Deep head pose: gaze-direction estimation in learning multimodal video", in Proceedings of the TMM, 17, 2015, pp. 2094–2107.
- [7] S. Qi, W. Wang, B. Jia, J. Shen, S.-C. Zhu, "Learning human-object interactions by graph parsing neural networks", in Proceedings of the ECCV, 2018, pp. 401–417.
- [8] Y. Lang, W. Liang, F. Xu, Y. Zhao, L.-F. Yu, "Synthesizing personalized training programs for improving driving habits via virtual reality", in Proceedings of the IEEE Conference on Virtual Reality, 2018.
- [9] C. Li, W. Liang, C. Quigley, Y. Zhao, L.-F. Yu, "Earthquake safety training through virtual drills", in Proceedings of the TVCG, 23(4), 2017, pp. 1275–1284.
- [10] W. Liang, J. Liu, Y. Lang, B. Ning, L.-F. Yu, "Functional workspace optimization via learning personal preferences from virtual experiences", in Proceedings of the TVCG, 25(5), 2019, pp. 1836–1845.
- [11] S. Sheikhi, J.-M. Odobez, "Combining dynamic head pose–gaze mapping with the robot conversational state for attention recognition in human-robot interactions", *Pattern Recognit. Lett.* 66 (2015) 81–90
- [12] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. learning-based, P. van der Smagt, D. Cremers, and T. Brox. "FlowNet: Learning optical flow with convolutional networks". ICCV, 2015.
- [13] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, R. Webb. "Learning from simulated and unsupervised images through adversarial training". In: CVPR 2017.
- [14] A. Kortylewski, B. Egger, A. Schneider, T. Gerig, A. Morel-Forster and T. Vetter, "Analyzing and Reducing the Damage of Dataset Bias to Face Recognition With Synthetic Data". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops pp. 0-0, 2019.
- [15] R. Queiroz, M. Cohen, J. L. Moreira, A. Braun, J. C. J. Júnior & S. R. Musse. "Generating facial ground truth with synthetic faces". In 2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images (pp. 25-31). IEEE, 2010.
- [16] A. Kortylewski, B. Egger, A. Schneider, T. Gerig, A. Morel-Forster & T. Vetter. "Analyzing and Reducing the Damage of Dataset Bias to Face Recognition With Synthetic Data". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 0-0), 2019.
- [17] Y. Wang, W. Liang, J. Shen, Y. Jia & L. F. Yu. "A deep Coarse-to-Fine network for head pose estimation from synthetic data". *Pattern Recognition*, 94, 196-206, 2019.
- [18] S. Bak, P. Carr, & J. F. Lalonde. "Domain adaptation through synthesis for unsupervised person re-identification". In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 189-205), 2018.
- [19] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev & C. Schmid. "Learning from synthetic humans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition" (pp. 109-117), 2017.
- [20] I. Alhashim, & P. Wonka. "High-Quality Monocular Depth Estimation via Transfer Learning". 1812.11941, 2018.
- [21] K. Lasinger, R. Ranftl, K. Schindler & V. Koltun. "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer". 2019.
- [22] W. Yin, Y. Liu, C. Shen, and Y. Yan, "Enforcing geometric constraints of virtual normal for depth prediction". In Proceedings of the IEEE International Conference on Computer Vision, pp. 5684–5693, 2019.
- [23] J. H. Lee, M. K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation". arXiv preprint arXiv:1907.10326, 2019.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/351575394>

Accurate 2D Facial Depth Models Derived from a 3D Synthetic Dataset

Conference Paper · January 2021

DOI: 10.1109/ICCE50685.2021.9427595

CITATIONS

2

READS

32

3 authors:



Faisal Khan

National University of Ireland, Galway

23 PUBLICATIONS 153 CITATIONS

SEE PROFILE



Shubhajt Basak

National University of Ireland, Galway

13 PUBLICATIONS 16 CITATIONS

SEE PROFILE



Peter Corcoran

National University of Ireland, Galway

624 PUBLICATIONS 4,625 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Data Augmentation using Generative Adversarial Networks (GAN) [View project](#)



Security and privacy for smartphones [View project](#)

Accurate 2D Facial Depth Models Derived from a 3D Synthetic Dataset

Faisal Khan
College of Engineering and Informatics
National University Ireland Galway
Galway, Ireland
f.khan4@nuigalway.ie

Shubhajit Basak
College of Engineering and Informatics
National University Ireland Galway
Galway, Ireland
s.basak1@nuigalway.ie

Peter Corcoran
College of Engineering and Informatics
National University Ireland Galway
Galway, Ireland
peter.corcoran@nuigalway.ie

Abstract— As Consumer Technologies (CT) seeks to engage and interact more closely with the end-user it becomes important to observe and analyze a user's interaction with CT devices and associated services. One of the most useful modes for monitoring a user is to analyze a real-time video stream of their face. Facial expressions, movements and biometrics all provide important information, but obtaining a calibrated input with 3D accuracy from a single camera requires accurate knowledge of the facial depth and distance of different features from the camera. In this paper, a method is proposed to generate synthetic high-accuracy human facial depth from synthetic 3D face models. The generated synthetic human facial dataset is then used in Convolutional Neural Networks (CNN's) for monocular depth facial estimation and the results of the experiments are presented.

Keywords—3D Facial models, Facial Depth models, CNN's

I. INTRODUCTION

Faces, with all their complications and an enormous number of degrees of freedom, allow us to connect and express ourselves through gestures, mimics and expressions. Depth information, pose, motion and shape are fundamental challenges in CT services and related devices. Examples include autonomous driving [1], license plate recognition [2], 3D reconstruction [3], scene understanding [4], human detection & pose estimation [5], and medical image segmentation [6]. Facial movements, biometrics and expressions all provide important information but obtaining accurate facial depth and distance of different features from the camera requires knowledge of the calibrated input with 3D information from a single camera. Nowadays, state-of-the-art structures rely on highly improved CNN's based designed networks and large datasets require high-power machines.

Progressively sophisticated camera hardware is becoming more reasonable at the consumer level, offering new possibilities. CT is now being combined with Machine Learning (ML) and Artificial Intelligence (AI) software to create new consumer-grade products. Luckily, recent advances in CT have taken to market numerous low-cost sensing solutions cameras can enable a range of useful CT applications including low-light facial recognition or object classification, business security and the world of home. Low-cost cameras can enable a range of useful CT applications including low-light

facial recognition or object classification, business security and the world of home, facial biometrics to authenticate users, portrait photography, classification of facial expressions (determine user emotion/mood), 3D models from the 2D camera (map face response onto a virtual reality (VR) avatar in an online world), TV (that can adjust the size of screen text or subtitles based on user-distance and preferences, 3D lighting effects, and demine head pose position and distance to optimize airbag deployment.

In particular, facial images are used in many CT structures. Facial images show various variations including expressions, 3D appearance, hairstyle and pose. The current advanced 3D tools such as Blender [7] and iClone [8] are used to synthesized many face variations. By using these 3D tools, large numbers of fake images can be created to train CNN's models. The generated images can be used for many applications having enough variations including depth, camera location and light direction and occlusions.

Deep learning-based networks require datasets having more information and precise data to train and evaluate different use cases methods for CT applications. In the past, years, researchers have made remarkable progress on 3D modelling and synthesis. Synthesized datasets have been used for deep learning models training in many tasks, example includes human behaviour analysis, driver monitoring, scene analysis and understanding, augmented reality systems, facial authentication and facial expression. The existing human facial datasets (e.g. Biwi Kinect Head Pose Dataset [9] and Pandora [10]) have lots of missing information especially the depth and due to the restricted variation, the number of available samples makes datasets insufficient for training deep learning models. These datasets required manual explanation of the scene that is very hard and time-consuming work and error-prone in case of videos [11]. In such type of facial data, they are not sufficient to learn well from CNN's model's limits many CT application [12-13].

Although, current deep learning-based methods have shown good performance on many tasks including face recognition systems, object classification, business security and the world of home, 3D reconstructions, robotics and autonomous driving. Purpose of accurate depth information in the 3D reconstruction is a very important part of computer vision problems. CT applications need more developments in short-

range depth estimation to engage with humans for better understanding.

In this paper, we proposed a details methodology for generating synthetic facial models. During the generation process, iClone [7] software and the 3D virtual human models are used to generate facial depth information. In the proposed method, by putting various variations in synthetic facial data we can produce any number of images, which require a more complex and detailed structure than the generative models used in the previous works.

II. LITERATURE REVIEW

Facial depth from monocular images as an ill-posed problem in computer vision, example includes virtual reality and human-computer interaction. Facial depth estimation is used in many applications including human object detection, human-robot interactions and controlling 3D avatars [14-19].

Recently, deep learning-based methods received a great interest in facial depth estimation, several works propose the use of RGB images with ground truth depth images to learn how to estimate depth [20-21]. The main issue is related to the available training datasets is limited size and overall low image quality [22-23].

Facial data is used for face recognition by expanding the real data for pose variation. Basel face model and 3D morphable [24-25] are used in many use cases applications to generate synthetic facial models [26]. A fake dataset is generated for person identification in [27]. (SURREAL) the dataset is proposed in [28], having a large number of synthetic images that are generated from 3D sequences of MoCap models. Fake human bodies are generated by using the SMPL model in [29] having a large number (6 million) frames with ground truth depth information, poses and mask segmentation. In this article, we present a methodology to create synthetic human facial models having various variations including camera location, light position, body-pose, facial animations and scene illuminations. The method can generate any number of images with ground truth depth information.

III. ORGANIZATION OF THE METHOD

In this section, we propose a complete pipeline for creating the synthetic human facial dataset with ground truth depth. Human facial models are generated by using the realistic human 100 models in iClone [7] and Blender [8] software in the following steps:

- The Initial human faces characters are generated by using the iClone character creator [7]. These animated facial models can be adapted with shaping and morphs in iClone character creator [7] which offers a useful 3D rigging option. An example of these models is shown in Fig 1.
- The synthetic human facial models are imported to iClone [7] with various expressions (happy, neutral, angry, scared and sad) to create more variation to the human facial models. An example is shown in Fig. 2.
- Synthetic human facial models have then exported to render high-quality images in different formats. The

generated human facial models are exported to Blender [8] from iClone [7] in .fbx format as it offers an appropriately rigging option. An example is given in Fig. 3.

- The human facial models were exported from iClone [7] and placed in a 3D scene in the Blender [8].
- The cameras and lights are placed in a fixed position and the relative distance of the model to the camera is changed within the range of 700-1000mm. The human facial model is rotated in the scenes and the sensor size is set between 36mm to 60mm. Fig. 4 show an example of the camera position and light location of the human facial models in Blender [8].
- During the generation process of the human faces with ground truth depth information, the (near and far) clip is set between 0.01 to 5 meters. RGB and depth images are generated in 480×640 resolution and texture, colour and static backgrounds. A few samples of the generated human facial models are shown in Fig. 5 while the camera location is varied to the corresponding human facial models.
- The position is changed at different points of the camera to the human facial models with the corresponding ground truth depth, which can be seen in Fig. 6.
- Blender [8] render passes are used to generate synthetic facial models. To reduce the noise, the branched path tracing method is utilized. An example is given in Fig. 7 of the noise controlling technique in Blender [8].
- Cycles engine are used to render the RGB and depth images, An example of the pipeline is given in Fig. 7, which show the generation procedure, camera position and light location.
- The generated synthetic human facial images with the ground-truth depth images are given in Fig. 9.
- In the last step, all the keyframes are rendered to get the RGB and the depth images are captured through the python plugin provided by Blender [8].

The whole experiments and human facial depth dataset creation is done on Core i7 with 32 GB of RAM and with GeForce Ti GTX GPU with (11x2) GB of the graphics card. The images are saved in .jpg and .exr format. The rendering average time for every frame is 52.5 seconds. The raw head pose and depth information are also taken as part of this human facial dataset. An example of the RGB and depth images with different head poses are presented in Fig 10. Different illuminations of the human facial dataset are shown in Fig 11. The more complex background is added to the human facial dataset and an example can be seen in Fig. 12.



Fig. 1. An example from the iClone Character creator.



Fig. 2. An example of Different expressions (happy, sad, angry, neutral and scared) of iClone [7].



Fig. 3. An example of iClone [7] facial model in Blender [8].

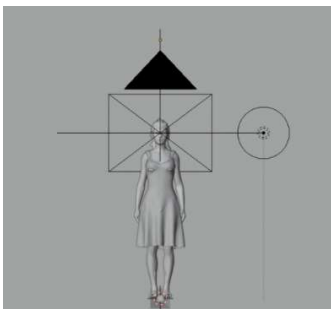


Fig. 4. An example of the 3D character in Blender [8] shows the light location and camera position.

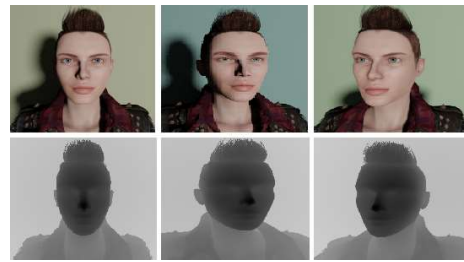


Fig. 5. An example of the head model from various views of the facial model and the corresponding depth information.



Fig. 6. Images of the synthetic human faces and corresponding ground truth depth in different camera location.

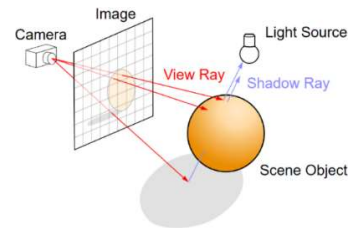


Fig. 7. An overview of the noise reduction method.

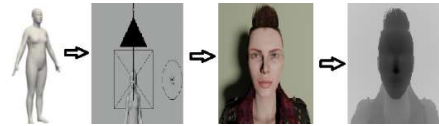


Fig. 8. A simple view of the rendering configuration in Blender [8].



Fig. 9. Human facial images and ground truth depth images with various expressions.



Fig. 10. An example of the facial images and their corresponding ground truth depth images with different head pose representation.



Fig. 11. An example of facial images with light variations.



Fig. 12. An example of the complex background representation of the facial images with ground truth depth.

IV. DEPTH ESTIMATION MODELS

A. Network architecture:

To check the data quality a shallow autoencoder (around 17 million parameters) with skip connection-based U-Net architecture shown in Fig 13 is proposed. The encoder and decoder both consist of basic blocks of double convolution with the Batch norm and ReLU activation. Additionally, in the decoder, the convolutions are used on the concatenation of the bilinear up-sampling of the earlier block with the corresponding block from the encoder module. The network has been initialized with random weight and trained with Adam Optimiser.

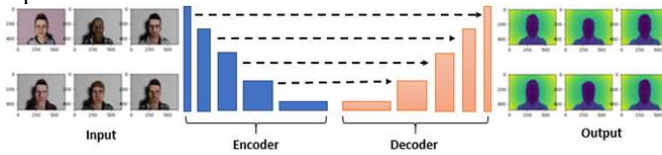


Fig. 13. An example of the proposed network architecture.

B. Training losses:

Loss function for monocular depth prediction from single image takes the difference among the ground truth g and the predicted depth map d . In this work, we have used SSIM loss, gradient loss and surface normal loss. These help to learn the correct depth of the scene as well as the 3D structure of the face. The loss L between g and d is defined as the weighted sum of the three different losses

$$L(g, d) = w_1 L_{SSIM}(g, d) + w_2 L_{grad}(g, d) + w_3 L_{SurfaceNorm}(g, d)$$

The first loss term L_{SSIM} incorporates the structural similarity (SSIM). As the SSIM has an upper bound value of one L_{SSIM} has been defined as follows

$$L_{SSIM}(y, \hat{y}) = \frac{1 - L_{SSIM}(g, d)}{Max\ Depth}$$

The second loss term L_{grad} is the L1 loss calculated over the image gradient of the depth image:

$$L_{grad}(g, d) = \frac{1}{n} \sum_p^n \nabla_x(e_p) + \nabla_y(e_p)$$

Where $\nabla x(e_p)$ denotes the spatial derivative of the difference of ground truth and predicted depth for p^h pixel e_p which stands for $(\|g_p - d_p\|)$ for the x-axis. The gradient of the depth maps has been obtained by the Sobel Filter and is sensitive to both x and y-axis. Though the gradient loss works well for strong edges it fails to penalise the small structural error like high-frequency undulation of a surface.

Lastly, to overcome the small structural errors, we used the $L_{SurfaceNorm}$ the loss which estimates the normal to the surface of the predicted depth map. The surface normal of the ground-truth and the predicted depth has been denoted as $n_p^g \equiv [-\nabla_x(g_p), -\nabla_y(g_p), 1]^T$ and $n_p^d \equiv [-\nabla_x(d_p), -\nabla_y(d_p), 1]^T$ and the loss has been calculated as the difference between the two surfaces normal:

$$L_{SurfaceNorm} = \frac{1}{n} \sum_p^n \left(1 - \frac{\langle n_p^d, n_p^g \rangle}{\|n_p^d\| \cdot \|n_p^g\|}\right)$$

Where $\langle ., . \rangle$ denotes the inner product of the vectors.

Additionally, as the loss term is larger where the ground truth depths are bigger, we used the reciprocal of the depth $[X, X]$. If the ground truth depth is y_{orig} we defined the target depth as $y = \frac{Max\ Dept}{y_{orig}}$.

We set the values of the weights w_1, w_2, w_3, w_4 as 0.1, 0.1, 0.1, 1 respectively.

C. Accuracy Measures:

To evaluate the result a commonly accepted evaluation method has been used with five evaluation indicators: Root Mean Square Error (RMSE), log Root Mean Square Error (RMSE (log)), Absolute Relative difference (AbsRel), and Square Relative error (SqRel), Accuracies. These are formulated as follows:

- $RMSE = \sqrt{\frac{1}{|N|} \sum_{i \in N} |d_i - g_i|^2}$
- $Average\ Log_{10}\ Error = \frac{1}{|N|} \sum_{i \in N} |\log(d_i) - \log(g_i)|$
- $Abs\ Rel = \frac{1}{|N|} \sum_{i \in N} \frac{|d_i - g_i|}{g_i}$

TABLE 1. RESULTS OF THE DEPTH ESTIMATION MODELS, SIMPLY U-NET, DENSEDEPTH [32] WITH VARIOUS BASE MODELS. FC REFERS TO THE FACIAL CROP WHICH MEANS THE ERRORS ARE ESTIMATED ONLY ON THE FACIAL REGION.

No.	Methods	AbsRel	SqRel	RMSE	RMSElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
1.	DenseDepth-161 [32]	0.0312	0.0121	0.0610	0.0169	0.9854	0.9876	0.9902
2.	DenseDepth-121 [32]	0.0320	0.0132	0.0712	0.0180	0.9732	0.9803	0.9880
3.	DenseDepth-169 [32]	0.0296	0.0096	0.0373	0.0129	0.9890	0.9920	0.9981
4.	DenseDepth-201 [32]	0.0375	0.0097	0.0304	0.0101	0.9920	0.9956	0.9969
5.	ResNet-101 [33]	0.0123	0.0210	0.0306	0.0089	0.9938	0.9965	0.9980
6.	ResNet-50 [33]	0.0232	0.0219	0.0445	0.0186	0.9919	0.9974	0.9984
7.	EfficientNet-B0 [34]	0.0145	0.0280	0.0360	0.0154	0.9912	0.9934	0.9978
8.	EfficientNet-B7 [34]	0.0132	0.0234	0.0353	0.0144	0.9880	0.9909	0.9965
9.	UNet-simple	0.0103	0.0207	0.0281	0.0089	0.9960	0.9976	0.9987
10.	UNet-simple (FC)	0.0098	0.0096	0.0143	0.0043	0.9982	0.9992	0.9996
11.	DenseDepth (FC)-169 [32]	0.0110	0.0074	0.0161	0.0034	0.9981	0.9990	0.9992
12.	ResNet (FC)-101 [32]	0.0132	0.0077	0.0170	0.0035	0.9980	0.9990	0.9992
13.	EfficientNet (FC)-B7 [34]	0.0112	0.0076	0.0166	0.0032	0.9887	0.9945	0.9989

^a. Results of the monocular depth estimation.

- Sq Rel = $\frac{1}{|N|} \sum_{i \in N} \frac{|d_i - g_i|^2}{g_i}$
- Accuracies = % of d_i s. t. $\max\left(\frac{d_i}{g_i}\right) = \delta < thr$

Where g_i is the ground truth and d_i is the predicted depth of the pixel i , N denotes the total number of pixels and thr denotes the threshold.

D. Experimentations

Table 1 shows the experimental results of the trained models on our datasets. Also, the depth has been masked within a certain range of 50 centimetres from the camera to evaluate the results only on the facial region of the images. We also used our synthetic human facial dataset and retrained state-of-the-art monocular depth estimation method [30] which is constructed on the encoder-decoder network with skip connections. A pre-trained DenseNet-169 [31] is used in the encoder, while in the decoder, a basic block of CNNs layers concatenated by a bilinear upsampling layer is used. Table 1, presents the results.

The encoder is replaced with several models while the decoder settings are unchanged. We tested with the technique using the synthetic human facial depth dataset, and provide the results in table 1.

In Table 1, the results of the simple U-Net based networks archive the best performance compared to the other networks on our generated synthetic human facial depth dataset. We study this as a result of the comparatively lower variance of the synthetic dataset as the models are only trained on a simple static background that leads to low-performance with big networks such as Dense Net, Res Net and efficient Net in this experiment. Also, we noted that the simple U-Net network-based encoder-decoder model holds

less than half the number of parameters and shows about two times faster compared to the other networks.

E. Implementations

We trained the network using the PyTorch. For training the model, we use adam optimizer for 20 epochs with 0.001 learning rate and batch size 6 on an NVIDIA 1080ti GPU's for all experiments. Fig. 14. Show the visual comparison of the methods presented in Table 1.



Fig. 14. An example of the qualitative comparison of methods. From left to right: Input, Ground Truth, U-Net, DenseDepth, ResNet and EfficientNet images.

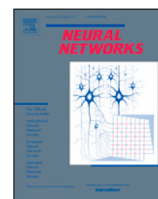
V. CONCLUSION

In this article, we present a method to generate synthetic facial depth dataset. The presented technique has a potential to create a large dataset of fake human facial images with ground depth information. The created synthetic human facial images can be used in many applications including 3D environments that will allow simulating real-life problems. Deep learning-based monocular depth estimation models are trained on the created facial dataset to validate the initial experiments that will further be extended to CT based

application with the focus on robotics, 3D reconstruction, beautification, autonomous vehicles, natural face modelling and augmented reality.

REFERENCES

- [1] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, A.M. Lopez, "The synthetic dataset: a large collection of synthetic images for semantic segmentation of urban scenes". in CVPR, 2016, pp. 3234–3243.
- [2] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, E. Magli, Robust license plate recognition using neural networks trained on synthetic images, *Pattern Recognit.* 93 (2019) 134–146.
- [3] H. Wang, J. Yang, W. Liang, X. Tong, Deep single-view 3d object reconstruction with visual hull embedding, in Proceedings of the AAAI, 2019.
- [4] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, R. Cipolla, Understanding real-world indoor scenes with synthetic data, in Proceedings of the CVPR, 2016, pp. 4077–4085.
- [5] L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, B. Schiele, Articulated people detection and pose estimation: Reshaping the future, in Proceedings of the CVPR, 2012, pp. 3178–3185.
- [6] I.K. Kallel, S. Almouahed, B. Solaiman, É. Bossé, An iterative possibilistic knowledge diffusion approach for blind medical image segmentation, *Pattern Recognit.* 78 (2018) 182–197.
- [7] 3D Animation Software: iClone: Reallusion. (n.d.). Retrieved from <https://www.reallusion.com/iclone/>.
- [8] Foundation, B. (n.d.). Home of the Blender project - Free and Open 3D Creation Software. Retrieved from <https://www.blender.org/>.
- [9] G. Fanelli, M. Dantone, J. Gall, A. Fossati, L. Van Gool, Random forests for real-time 3d face analysis, in Proceedings of the IJCV, 101, 2013, pp. 437–458.
- [10] G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara, "Poseidon: Face-from-depth for driver pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4661–4670.
- [11] T. List, J. Bins, J. Vazquez, & R. B. Fisher. Performance evaluating the evaluator. In 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance. 2005, pp. 129-136. IEEE.
- [12] S. R. Musse, R. Rodrigues, M. Paravisi, J. C. S. Jacques. Junior, and C. R. Jung. "Using synthetic ground truth data to evaluate computer vision techniques". In IEEE Workshop on Performance Evaluation of Tracking Systems (in conjunction with ICCV 07), pages 25–32, 2007.
- [13] G. R. Taylor, A. J. Chosak, and P. C. Brewer. Ovvv: "Using virtual worlds to design and evaluate surveillance systems". In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.
- [14] S.S. Mukherjee, N.M. Robertson, "Deep head pose: gaze-direction estimation in learning multimodal video", in Proceedings of the TMM, 17, 2015, pp. 2094–2107.
- [15] S. Qi, W. Wang, B. Jia, J. Shen, S.-C. Zhu, "Learning human-object interactions by graph parsing neural networks", in Proceedings of the ECCV, 2018, pp. 401–417.
- [16] Y. Lang, W. Liang, F. Xu, Y. Zhao, L.-F. Yu, "Synthesizing personalized training programs for improving driving habits via virtual reality", in Proceedings of the IEEE Conference on Virtual Reality, 2018.
- [17] C. Li, W. Liang, C. Quigley, Y. Zhao, L.-F. Yu, "Earthquake safety training through virtual drills", in Proceedings of the TVCG, 23(4), 2017, pp. 1275–1284.
- [18] W. Liang, J. Liu, y. Lang, B. Ning, L.-F. Yu, "Functional workspace optimization via learning personal preferences from virtual experiences", in Proceedings of the TVCG, 25(5), 2019, pp. 1836–1845.
- [19] S. Sheikhi, J.-M. Odobez, "Combining dynamic head pose-gaze mapping with the robot conversational state for attention recognition in human-robot interactions", *Pattern Recognit. Lett.* 66 (2015) 81–90.
- [20] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE International Conference on Computer Vision, pages 2650–2658, 2015.
- [21] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In Advances in neural information processing systems, pages 2366–2374, 2014.
- [22] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In 3D Vision (3DV), 2016 Fourth International Conference on, pages 239–248. IEEE, 2016.
- [23] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1119–1127, 2015.
- [24] T. Shrivastava, O. Pfister, J. Tuzel, W. Susskind, R. Wang, Webb. "Learning from simulated and unsupervised images through adversarial training". In: CVPR 2017.
- [25] A. Kortylewski, B. Egger, A. Schneider, T. Gerig, A. Morel-Forster and Vetter, T, "Analyzing and Reducing the Damage of Dataset Bias to Face Recognition With Synthetic Data". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops pp. 0-0, 2019.
- [26] R. Queiroz, M. Cohen, J. L. Moreira, A. Braun, J. C. J. Júnior & S. R. Musse. "Generating facial ground truth with synthetic faces". In 2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images (pp. 25-31). IEEE, 2010.
- [27] Y. Wang, W. Liang, J. Shen, Y. Jia & L. F. Yu. "A deep Coarse-to-Fine network for head pose estimation from synthetic data". *Pattern Recognition*, 94, 196-206, 2019.
- [28] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev & C. Schmid. "Learning from synthetic humans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition" (pp. 109-117), 2017.
- [29] S. Bak, P. Carr, & J. F. Lalonde. "Domain adaptation through synthesis for unsupervised person re-identification". In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 189-205), 2018.
- [30] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.
- [31] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261–2269, 2017.
- [32] I. Alhashim, & P. Wonka. "High-Quality Monocular Depth Estimation via Transfer Learning". 1812.11941, 2018.
- [33] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
- [34] Tan, M., & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:190*.



An efficient encoder–decoder model for portrait depth estimation from single images trained on pixel-accurate synthetic data[☆]



Faisal Khan^{a,*}, Shahid Hussain^b, Shubhajit Basak^c, Joseph Lemley^d, Peter Corcoran^a

^a Department of Electronic Engineering, College of Science and Engineering, National University of Ireland Galway, Galway, H91 TK33, Ireland

^b Data Science Institute, National University of Ireland Galway, Galway H91 TK33, Ireland

^c School of Computer Science, National University of Ireland Galway, Galway H91 TK33, Ireland

^d Xperi Corporation, Block 5 Parkmore East Business Park, Galway, H91V0TX, Ireland

ARTICLE INFO

Article history:

Received 11 April 2021

Received in revised form 13 June 2021

Accepted 5 July 2021

Available online 13 July 2021

Keywords:

Depth estimation

Facial depth

2.5D dataset

Hybrid loss function

Convolution neural network

Encoder–decoder architecture

ABSTRACT

Depth estimation from a single image frame is a fundamental challenge in computer vision, with many applications such as augmented reality, action recognition, image understanding, and autonomous driving. Large and diverse training sets are required for accurate depth estimation from a single image frame. Due to challenges in obtaining dense ground-truth depth, a new 3D pipeline of 100 synthetic virtual human models is presented to generate multiple 2D facial images and corresponding ground truth depth data, allowing complete control over image variations. To validate the synthetic facial depth data, we propose an evaluation of state-of-the-art depth estimation algorithms based on single image frames on the generated synthetic dataset. Furthermore, an improved encoder–decoder based neural network is presented. This network is computationally efficient and shows better performance than current state-of-the-art when tested and evaluated across 4 public datasets. Our training methodology relies on the use of synthetic data samples which provides a more reliable ground truth for depth estimation. Additionally, using a combination of appropriate loss functions leads to improved performance than the current state-of-the-art network performances. Our approach clearly outperforms competing methods across different test datasets, setting a new state-of-the-art for facial depth estimation from synthetic data.

© 2021 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The problem of estimating depth from the image data of a scene is a fundamental task in computer vision. It is particularly important in *image understanding* where it is desirable to determine the primary objects and regions within an imaged scene and where their relative locations and orientations from frame-to-frame can provide valuable information about scene activity. While single frame object detection (Chang & Wetzstein, 2019) and classification techniques (Athira & Khan, 2020) are quite well advanced depth estimation is typically a more challenging problem (Fan et al., 2021).

The classic approach to depth estimation is to employ a two-camera, stereoscopic solution, mimicking the human visual system, and using disparity between the two images to construct a

depth map (Wenxian, 2010). When camera motion is available, or when objects move from frame-to-frame it is possible to use this data to reconstruct depth maps for individual image frames, especially in mobile or handheld devices which incorporate modern inertial motion sensing (Schöps, Sattler, Häne, & Pollefeys, 2017). However there are applications where only a single camera is used and exact motion sensing is not available and thus it is desirable to estimate a depth map of an imaged scene from single image frames. The current work is focused on this task, and in particular in understanding if it is feasible to improve on current state-of-the-art (SoA) while reducing the complexity of the computational model.

Human faces are one of the most common objects found in images and an important component of many *image understanding* problems. It is well-known from human anthropometry that the eye-separation in a human face falls into a narrow range (Ware, 2019) and thus given a knowledge of the field-of-view of a camera it is possible to determine with reasonable accuracy the distance-to-camera of a human subject from a single image frame. This research work speculates that it should be feasible to train a neural computer vision model to learn a more accurate depth estimation by training it on data that includes

[☆] This work was supported by the College of Science and Engineering, National University of Ireland Galway, Galway, H91TK33 Ireland; the Xperi Galway Block 5 Parkmore East Business Park, Galway, H91V0TX, Ireland; and the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224.

* Corresponding author.

E-mail address: f.khan4@nuigalway.ie (F. Khan).

human faces. With sufficient data and a pixel-accurate ground truth (GT) the model should learn many nuances of human facial features and structure that can improve depth estimation over current SoA.

The main contribution of this work is an improved, deep learning based encoder–decoder model for depth estimation from single image frames. This model is more computationally efficient than current SoA depth estimation models and shows performance equal to, or better than SoA when evaluated across 4 public datasets. In part this improved performance is achieved through our training methodology which relies on the use of synthetic data samples that can provide a more accurate GT for depth than is available from existing public datasets. Details of this synthetic training dataset and the associated training methodology provide a second significant contribution of this work.

The rest of this paper is organized as follows. Section 2 presents a review of the related (depth estimation) literature while the details of the synthetic human facial dataset used in our training methodology are presented in Section 3. The evaluation methodology of the compared methods is described in Section 4. Section 5 provides details of the encoder–decoder model and the associated loss functions used in the training process. A rich synthetic human facial dataset is employed in the training process as described and details of a series of experimental comparisons of our model with current SoA models for depth estimation are outlined in Section 6. Finally a discussion of the outcomes of this research work is briefly discussed in Section 7 and the potential for future refinement and improvements is provided in Section 8.

2. Related works

Depth estimation is the method of preserving 3D information of a scene using 2D information captured by cameras. Monocular depth estimation, also known as depth estimation from a single image (DESI), is achieved by using only one image. These techniques are designed to estimate distances between scene objects from a single point of view. This necessitates using these methods on low-cost embedded systems for performance estimation.

There has been a significant improvement in DESI methods over the past couple of years (Basha, Avidan, Hornung, & Matusik, 2012; Javidnia & Corcoran, 2017; Laidlow, Czarnowski, & Leutenegger, 2019; Ranftl, Lasinger, Hafner, Schindler, & Koltun, 2020; Tian & Hu, 2021). Most of the deep learning-based methods involve a CNN trained on RGB images and the corresponding depth maps. These methods can be categorized into supervised, semi-supervised, and unsupervised. A brief literature review based on deep learning monocular depth estimation methods can be found in Khan, Salahuddin, and Javidnia (2020).

Supervised DESI techniques use an input image and the corresponding depth maps for training. In such a case, the trained network can directly output the depth predication (Yin, Liu, Shen, & Yan, 2019). Supervised deep learning approaches have achieved SoA performance in the DESI task (Andraghetti et al., 2019; Chen, Zhao, Hu, & Peng, 2021; Fu, Gong, Wang, Batmanghelich, & Tao, 2018; Goldman, Hassner, & Avidan, 2019; Lee, Han, Ko, & Suh, 2019; dos Santos Rosa, Guizilini, & Grassi, 2019; Wang et al., 2020). Despite the fact that these methods can predict accurate depth maps when testing on the same or similar datasets, they do not generalize well to scenes beyond the original dataset (Ranftl et al., 2020). Also, the performance of these supervised methods required a large amount of high-quality depth data and thereby are unable to generalize to all use cases.

To overcome the need for high-quality depth estimation as seed data, many methods have been employed to train the depth estimation network in a semi-supervised manner. Numerous

semi-supervised methods are proposed, which require smaller amount of labeled data and large amount of unlabeled data for training (Bazrafkan, Hossein, Joseph, & Corcoran, 2017; Choi et al., 2020; Lei, Wang, Li, & Yang, 2021; Yue, Fu, Wu, & Wang, 2020; Yusionsg & Naval, 2020; Zhao, Jin, Wang, & Wang, 2020). Semi-supervised methods, on the other hand, suffer from their biases with more information is required, such as sensor data and camera focal length (Xian et al., 2020).

To train the networks for depth estimation, self-supervised methods only require a small number of unlabeled images (Yusionsg & Naval, 2020). Many tasks have been studied using self-supervised methods, including 3D reconstruction (Wang, Yang, Liang, & Tong, 2019), human detection and pose estimation in DESI (Guizilini, Ambrus, Pillai, Raventos, & Gaidon, 2020; Johnston & Carneiro, 2020; Klingner, Termöhlen, Mikolajczyk, & Fingscheidt, 2020; Li et al., 2021; Poggi, Aleotti, Tosi, & Mattoccia, 2020; Spencer, Bowden, & Hadfield, 2020; Widya et al., 2021). These methods automatically obtain depth information by correlating various image input modalities. However, self-supervised methods suffer from generalization issues. The models can only perform on a very limited set of scenarios with distributions similar to the training set.

We argue that high-quality deep learning-based DESI methods can in principle operate on a fairly wide and unconstrained range of scenes. What limits their performance is the lack of large-scale, dense GT that spans such a wide range of conditions (Ranftl et al., 2020). Several of the existing benchmark datasets: Pandora (Borghi, Venturelli, Vezzani, & Cucchiara, 2017); Eurecom Kinect Face (Min, Kose, & Dugelay, 2014); Biwi Kinect Head Pose (Fanelli, Weise, Gall, & Van Gool, 2011) have been tested with limited sample sizes (250k, 50k and 15k) and fewer variations to estimate around 24, 52, and 20 subjects. It can be noted in particular that these datasets show only a small number of dynamic objects. Networks that are trained on data with such strong biases are prone to fail in less constrained environments (Xian et al., 2020).

Despite their capacity to provide the depth layout without any domain knowledge, deep learning-based techniques still struggle with inconsistencies at the depth boundary. Existing approaches, in particular, rely on characteristics taken from well-known encoders. The decoding mechanism in the symmetric design simply upsamples these latent features to their original size, and then converts them into the depth map. Because this translation procedure struggles to incorporate object depth boundaries at multiple scale levels, it is likely to produce inaccurate depth values between object boundaries. A unique yet simple method for monocular depth estimation was developed to address the shortcomings of prior approaches. The suggested method's main idea is to use the Laplacian pyramid-based decoder architecture to correctly interpret the relationship between encoded characteristics and the final output for monocular depth estimation (Song, Lim and Kim, 2021).

A new method called dense prediction transformer (DPT) is introduced. It is a dense prediction architecture based on an encoder–decoder design that uses a transformer as the encoder's primary computational building block. It also has a global receptive field at every level, demonstrating that these qualities are particularly beneficial for dense prediction problems because they naturally result in fine-grained and globally coherent predictions (Ranftl, Bochkovskiy, & Koltun, 2021). An investigation of a method in which the network learns to focus adaptively on depth range regions that are more likely to occur in the scene of the input image for depth estimation (Bhat, Alhashim, & Wonka, 2020). To create per-pixel depth maps with sharper bounds and richer depth features, a novel framework called MLDA-Net is proposed. A multi-level feature extraction (MLFE) technique that can

learn rich hierarchical representation and to amplify the obtained features both worldwide and locally, a dual-attention technique combining global and structure attention is developed, resulting in better depth maps with sharper borders (Song et al., 2021).

CoMoDA is a new self-supervised Continuous Monocular Depth Adaptation approach that adapts the pretrained model on the fly on a test video. Rather than using isolated frame triplets as in conventional test-time refinement methods, they choose for continuous adaptation, which relies on earlier experience from the same scene (Kuznietsov, Proesmans, & Van Gool, 2021). To reduce inaccurate inference of depth details and the loss of spatial information, a new detail-preserving network (DPNet), which is a dual-branch network architecture that fully overcomes the aforesaid issues and makes depth map inference easier (Ye, Chen, & Xu, 2021).

To improve the training efficiency of deep neural networks, more accurate labeled synthetic human facial image datasets could be used. The synthetic datasets can be created by a camera using sensing technologies or by using available software tools, which are less expensive, require less effort, and produce better face models that resemble a realistic 3D environment (Koo & Lam, 2008; Roy-Chowdhury & Chellappa, 2005). During the training process, the weight adjustment at each node through the activation functions are controlled according to the efficiency of the loss functions and thereby the use of appropriate loss functions further improves the performance of the deep neural networks (Jiang, El-Shazly, & Zhang, 2019; Lee & Kim, 2020; Liu, Zhang, Meng, & Gao, 2020). The use of synthetic datasets and the selection of appropriate training methodology can help in the human facial depth estimation. Overall, none of the current datasets is large enough to support the development of a model that can reliably work on real images from a wide range of scenes. Currently, we are confronted with a number of datasets that may be useful when combined, but are individually biased and incomplete.

3. Modeling of the synthetic dataset

This section presents a detailed pipeline of creating the synthetic dataset. Most of the datasets currently available for facial depth estimation have a very limited amount of ground truth (GT) which makes them unsuitable for training deep learning models (Borghi et al., 2017; Fanelli et al., 2011; Min et al., 2014). Besides, due to practical limitations in data acquisition, most of the depth GT are error-prone. Datasets with multiple facial pose representations are especially prone to errors in the depth GT data.

Furthermore, the acquisition of facial data from subjects is now subject to a range of privacy regulations and ethical constraints. In Europe the General Data Protection and Regulations (GDPR) govern the acquisition and distribution of personal data introducing new challenges for researchers working with data from live humans. This makes a case for generating inexpensive synthetic dataset with lower complexity and a rich amount of labeled data resembling the features of realistic human models such as the camera parameters, positions, light locations, scene illuminations and other constraints within a 3D environment.

This work introduces a methodology to build synthetic human facial datasets. This methodology leverages a commercial tool for generating synthetic avatars, iClone and Character Creator (CC) employs an open access 3D animation environment, Blender to build a rich variety of scenes for rendering 2D data samples with matching, pixel exact, depth GT. Once avatar models are exported into the 3D environment it is relatively straight forward to vary the rendering camera location and positions, camera model and acquisition parameters together with controlling the scene

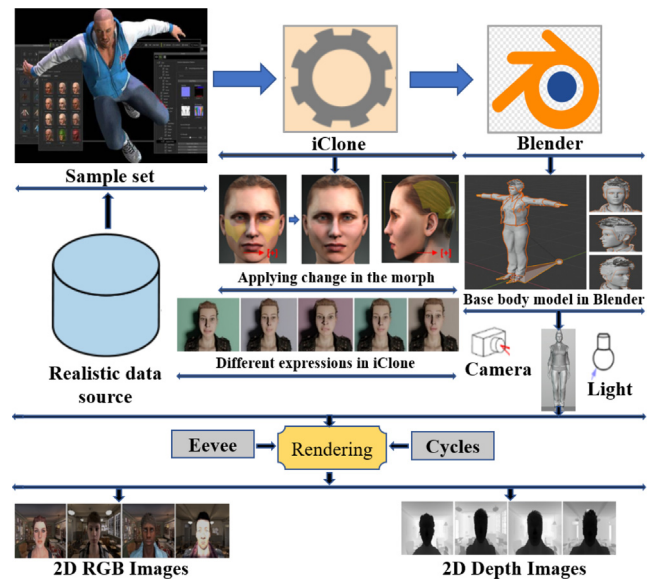


Fig. 1. A schematic representation of generating the synthetic human facial dataset: Samples from the 100 Realistic Head Models, with variation in gender, race, and age. In iClone, changing the morph to create variations to the head models. Importing fully rigged FBX models from iClone to Blender, lighting, camera positioning, and generating the final 2D images.

backgrounds, lighting sources, and absolute head pose. Facial animations can also be used and variations in facial expression can be introduced. Most importantly, all of the inputs to build a particular 3D scene can be recorded and reproduced exactly in a way that is not feasible for a real-world data acquisition.

Naturally, synthetic facial data will not have the same richness in terms of skin features as real image data. But given the other benefits of using synthetic data to train a neural DESI model, a key research question that we seek to answer in this work is whether we can achieve comparable accuracy to SoA DESI models that are trained on real-world data?

Our procedure for generating the synthetic dataset is illustrated in Fig. 1 and the detailed description is presented in the subsections.

3.1. Synthetic human model with 3D scene setup

Previous works (Elanattil & Moghadam, 2019; Gu, Yang, De Mello, & Kautz, 2017; Varol et al., 2017) with synthetic virtual humans relied on high-quality 3D scans to produce synthetic data from 3D human models. But these 3D scans are expensive and difficult to capture due to different data regulation laws like GDPR, so there is a very limited number of variations in the currently available synthetic facial depth datasets. This study uses the low-cost commercially available 3D asset creation software and an open-source 3D computer graphics (CG) tool as an alternative to creating virtual human models. Fig. 2 shows an example of these models.

3.1.1. The iClone character creation process

The characterization of virtual human models is achieved with realistic human faces, humanoid behaviors, and 3D riggings through the iClone CC process. In the process the template is applied to the base body while the sculpting and morphs features are utilized for capturing the facial animations. A realistic facial expressions and morph transformation are then applied in the 3D mesh that enhance the variations in the data. The virtual human face models are imported from CC to iClone.



Fig. 2. From left to right: Samples from the 100 Realistic Human Models with variation in gender, race, age and facial expressions followed with a fully rigged FBX model from iClone to Blender with the mesh representation.

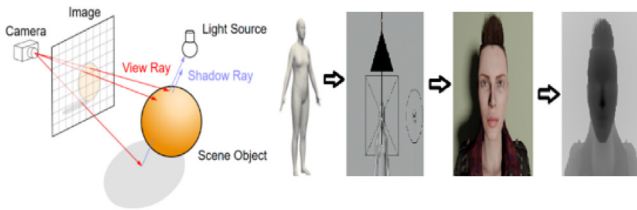


Fig. 3. In Blender, a simplified view of the rendering configuration. The left row shows the body shape, light and camera setting information; the middle row shows the facial RGB image and the last row illustrates the corresponding facial depth image.

3.1.2. Adding variations to models in iClone

The iClone provides a rich features library with embedded templates supporting full parameter control for shapes, textures, clothes materials modification and representations in different styles. The layout base is easily adjustable to all the sub-nodes by rotating them through different angles from hair element to the coordinates texture and facial expressions. Such features are implemented to specify the models with a range of human characteristics including neutral, angry, happy, sad, and scared along with the customized fabric plates layers and five different colored hairstyle that results in generating above hundred variations for the facial model.

3.1.3. Model transfer from iClone to Blender

To capture a richer GT with dense facial depth, head pose, camera locations, scene illuminations the model needs a transformation interface from iClone to Blender software. The interface is designed by coupling the 3D modeling software to adjust the adaptation of FBX format between the different software tools.

3.1.4. Manipulating models in Blender

Blender is a 3D creation suite open-source tool that provides full support for modeling, rigging, animation, simulation, rendering, composition, motion tracking, video editing and game creation (with python integration) over the entire 3D model. The rigs animations are controlled with the constraint keyframes and shape keys, while the camera parameters are configured by adjusting the field of view (FOV), the clip zoom in–out values, sensors size, depth field and the f-stop values. Furthermore, the light paths of refraction, reflection, diffraction, and absorption are tracked through realistic cycle rendering engine as illustrated in Fig. 3.

3.1.5. Building 3D scenes in Blender

The FBX format alignment allowed us to control and adjust the head motions of various angles, while illuminations such as area, sun, point, and spotlight assisted in varying the lights based on the realistic scenarios of the scene. The GT rendering of the image is achieved through admission of the camera model to the particular scene mode, during the cycle rendering engine control process. The ground truth data is generated by conducting

a sequences of head movements experiments through controlling the neck bone rotations over the FBX based model. In the process the initial head position is maintained by scaling an arbitrary object between the eyeballs under the range of the camera focal point.

The translation and the rotations of neck bones are transferred to the arbitrary object in a way by retaining the constraints of the original object. The default setting of Blender does not allow the head to be positioned at zero angle therefore the imported model head moment is restricted by default. The initialization of head frame position is performed by setting down the yaw, pitch and roll of the initial frame in the Blender world coordinator, the original neck bone is then rotated by wisely minimizing the delta through a python script, that tuned the local coordinates x, y, and z-axis of arbitrary object to zeros. After the initial setup, a sequential (Pitch, roll, and yaw) uniform rotation was applied to the neck bone and a balanced status of all the frames was recorded. The yaw, pitch and roll of the head pose are calculated by capturing the corresponding values from the rotation matrix. The ranges of the yaw, pitch, and roll have been maintained in range of $\pm 80^\circ$, $\pm 70^\circ$ and $\pm 55^\circ$, respectively, with the granularity of 3° angle.

3.1.6. The Blender camera model

The Blender camera specifies the lens focal length and aperture parameters for defining the viewpoint of the scenes and their rendering. The default camera model is applied to the scene, and its properties are adjusted to replicate the real environment. The camera is set at 30 centimeters distance from nose tip of the model and the background plane is set at a distance of 2 m, respectively. The camera sensors size and FOV are set at 36 millimeters (mm) with 60° and the near and far clip are set at 0.001 and 5.0 meters (m), which results in covering the overall scenes. The representation of 3D objects with 2D images is obtained through optimizing the camera lens options. The camera placement was maintained at a fixed position while the human model was placed within the range of 700–1000 mm relative to the camera that replicate the capturing of data in realistic scenarios. Finally, the realistic 2D images are obtained by a random selection of main camera translation, head camera translation and rotations.

3.1.7. 3D background scene selections in Blender

A mix of plain, textured, and real images have been used to add variations to the background. The background of the scene was varied to provide more variations in order to improve model generalization. The Brodatz-based color images provided by Abdelmounaime and Dong-Chen (2013) are used for the textured background. The classroom and barbershop scene from Blender Eevee were chosen for the complex background.

3.1.8. Ground truth rendering in Blender

Blender provides Cycles and Eevee render engines for path tracing and rasterization functions, respectively. To obtain a realistic rendering, the Cycles rendering engine is used as cycles is Blender most feature-rich and production-proven renderer. The path tracers function captures the light reflection, refraction, and adsorption while the rasterization maintained the pixel information for a fast rendering process but reduced the accuracy. It has been observed that the degrade in accuracy is due to the rendering process of transparent materials and noises during their Cycle path tracing. The noises are reduced by the branched path tracing mechanism, which splits the original ray by capturing its reflected rays in multiple directions that provide a full control over the shades and support the accuracy improvement.



Fig. 4. Random sample frames with high-resolutions RGB images and their corresponding ground truth depth with different variations (head poses, expressions, light variations, camera positions, clothes, viewpoints and backgrounds: plain; textured; real) obtained from the generated synthetic dataset.

The movement of most of the other parts of body are controlled according to the structures of their bones. The RGB render pass was used in the Blender compositor setup to get the final render. The head and the shoulders bone are identified in the pose mode then the head mesh is rotated with respect to the selected bones and the selected key frames are recorded. Finally, all the key frames are rendered by capturing their respective head poses through the python plugins and the RGB and the depth images are obtained.

3.2. Dataset information

Following the methodology outlined above, the proposed framework works as follows: In CC, a set of virtual human models is constructed using the Real 100 humans face models. To add more variation, the texture and morphology of the models are changed. These models are then sent to iClone, where different facial expressions are imposed. The mesh, textures, and animation keyframes for the final 3D models with facial expressions are exported in FBX format. Complete information can be found in Sections 3.1.1 and 3.1.2.

Following that, the FBX files are imported and scaled in Blender world coordinate system. Lights and cameras are added to the scene, and their properties are adjusted to capture the real environment. The render layer RGB and Z-pass outputs are then set up in the compositor to get the final result. In pose mode, the head and shoulder bones are identified, and the head mesh is rotated in relation to those bones, with the keyframes saved. Finally, all of the keyframes are rendered to obtain RGB and depth images, and the appropriate head pose (yaw, pitch, and roll) is captured using Blender Python plugin. Sections 3.1.3–3.1.8 contain the detailed information. GT is rendered on an Intel Core i5-7400 3 GHz CPU with 32 GB RAM and an NVIDIA GeForce GTX TITAN X Graphical Processing Unit (GPU) with 24 GB of dedicated graphics memory.

For each frame, the RGB images are rendered with 640×480 resolutions and saved in jpg format and the corresponding depth data is saved in a raw file (.exr format). Additionally, the head

pose information for each frame is captured and saved in a text (.txt) file. Cycle Rendering Engine, Blender physically-based path tracer for production rendering, took an average of 26.3 s to render each 2D image frame. The total dataset size is around 3500k image samples, with approximately 3.5k 2D image samples per subject. For each of the 100 face models, the data is saved in its own folder. The rendered RGB images and the corresponding Gt (depth and head pose) for each face model are stored in three different paths for the three types of backgrounds – simple, textured, and complex. The sample frames with their ground truth depth images and different backgrounds (simple, textured and complex) obtained from the synthetic dataset are illustrated in Fig. 4.

The generated synthetic dataset used in this research work consists of 3D virtual human models and 2D rendered RGB and GT depth images in zipped version with a total size of 650 GB categorized into two folders. All of the CC and iClone data information (textures, .fbx, .fbm, and .blend) for each subject is contained in the 3D virtual models folder, which is further divided into sub-folders (male, female). The male and female sub-folders of the 2D rendered images folder contain 56 and 44 subjects, respectively. For the three types of backgrounds – simple, textured, and complex – these subjects are stored in three different paths. The sample and texture path are divided into five main directories (happy, sad, neutral, scared, and angry), each of which contains the RGB images, depth images, and raw head pose data for each frame. The complex directory is divided into two main folders, classroom and barbershop, which have the same structure as the sample and textured folders. The file hierarchy structure is shown in Fig. 5.

Our synthetic dataset¹ is available for a free of cost download and can be utilized for scientific research purposes.

In contrast to the existing datasets (Borghetti et al., 2017; Fanelli et al., 2011; Min et al., 2014) our dataset provides a richer set of portrait scene detail. Examples include a pixel-exact GT depth information corresponding to each rendered RGB image; a larger

¹ https://github.com/khan9048/Facial_depth_estimation.

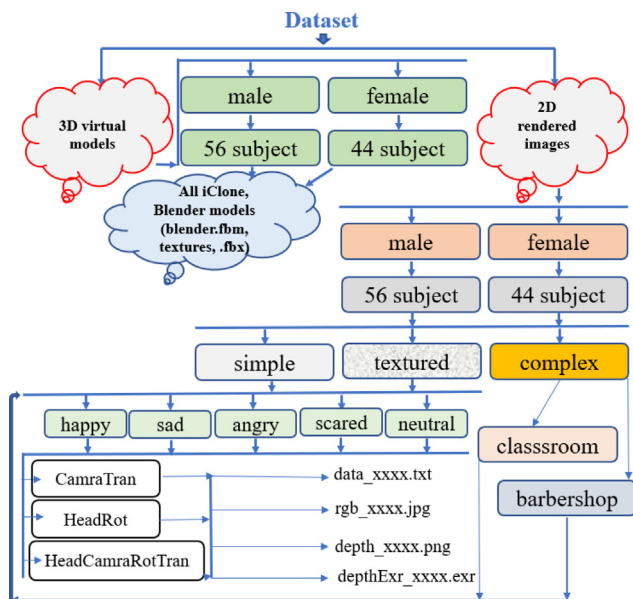


Fig. 5. Dataset organization: The dataset is divided into different folders which correspond to each 'subject' being captured and rendered with RGB images; ground truth depth images.

number of training samples; variations in camera perspective, facial expression and head pose. Most importantly each 3D scene data can be exactly replicated, and new variations introduced to test the importance of different elements of scene composition.

4. Evaluating state-of-art models for single image depth estimation

The purpose of this study is to see how well synthetic facial depth data can be used to estimate facial depth estimation. A set of SoA DESI neural networks is used to analyze the generated synthetic human facial depth dataset. Since there are no publicly available benchmarks methods for the evaluations purposes, this work used DESI neural networks to train over the generated synthetic dataset and evaluate with test data. In addition, a new CNN model is proposed, and its performance is evaluated against the SoA networks. Initially, SoA DESI methods BTS (Lee et al., 2019), Densedept (Alhashim & Wonka, 2018) and UNet-simple (Khan, Basak, & Corcoran, 2021) are trained using the synthetic human facial dataset and the results are compared against the proposed network.

The most important requirement for a sensible training scheme is that computations are performed in an appropriate output space that is compatible with all GT representations. As a result, the GT was scaled to the generated dataset for training the SoA methods. A typical CNN system comprises of certain layers which include convolution layers, pooling layers, dense layers, and fully connected layers. There are a variety of pre-trained networks that can be used to perform tasks like visual recognition, object detection, segmentation, and depth estimation. This work employ a pool of pre-trained networks which includes EfficientNet-B0, EfficientNet-B7, ResNet-101, ResNet-50, DenseNet-169, DenseNet-201, DenseNet-161 to generalize the model for the target facial depth estimation.

Although these methods can produce depth maps with comparable accuracy, they are computationally more expensive and requires large amount of graphical memory. As an alternative, the proposed model in this work automates the collection of optimal parameters, thus reducing model complexity during the training

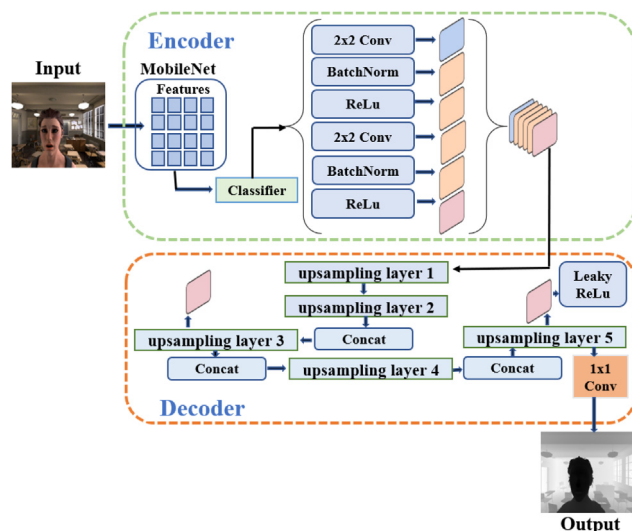


Fig. 6. Schematic diagram of the proposed depth estimation network: A multi-layer Encoder–Decoder network is used to generate accurate facial depth maps based on the MobileNet backbone model.

process, and is more computationally efficient than the current SoA depth estimation models and shows performance equal to, or better than SoA when tested across 4 public datasets.

We examine how to compare the effects of various methods for estimating a scene facial depth from a single image frame. A new evaluation protocol of SoA facial depth estimation algorithms for synthetic dataset is proposed, setting up a new SoA for facial depth estimation.

Section 5 provides details of the Encoder–decoder model and the associated loss functions used in the training process. In Section 6, we present a detailed analysis of our model performance against these methods using four public datasets. Also, a brief comparison analysis, evaluation matrices, test datasets, implementation details, encoders comparison and qualitative study are presented.

5. An encoder–decoder based facial depth estimation model

In this section, we described the proposed single image depth estimation network with encoder–decoder mechanism and hybrid loss function to optimally select the hyper parameters for improving the training process over the generated synthetic dataset.

5.1. Network architecture

To analyze the validity of the generated datasets, a CNN network is designed that is referred to as FaceDepth and its performance is compared against the SoA architectures. A schematic diagram of the proposed model is illustrated in Fig. 6. It consist of input and output images and a detailed Encoder–decoder network architecture. The Encoder–decoder learn to map data-points from an input domain to an output domain via a two-stage mechanism in the network. In the first stage the encoder function $f = f(x)$, compresses the input into a latent-space representation while in the second stage the decoder function $y = g(f)$ predicts the output. In the encoder, we employ MobileNet (Sifre & Mallat, 2014) which is based on depthwise decomposition process to factorize the CNN layers into depthwise and pointwise layers. Each of the depthwise layers utilize the filtration function that extracts low-resolution features from the input image. The extracts

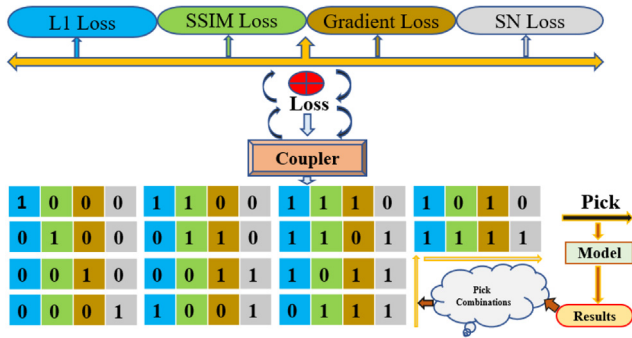


Fig. 7. An illustration of the hybrid loss function composition: A hybrid loss function is introduced through the combination of point-wise loss, gradient loss, surface normal loss, and SSIM loss functions.

features are then fed to the decoder, which refines, merge and upsample them to the final high-resolution output depth map. In the second stage of the network, the decoder consists of five upsampling and a single pointwise layers. Each upsampling layer performs a 5x5 CNN and reduces the number of channels with a ratio of 2:1 input and output channels. Three skip connections are applied to reconstruct a more detailed dense information for the final depth map. The hybrid loss function measures the differences between the GT depth and the predicted depth map to minimize the reconstruction errors. A detailed description of the hybrid loss function is presented in the subsequent section.

5.2. Hybrid loss function

The loss functions estimate the image depth by measuring the difference between the true depth (g) and predicted depth (d) such that the loss function results in a higher error if d deviates largely from g and vice versa. To fine-tune and to penalize the distortion among the GT and predicted depths for high frequency images a hybrid loss function is introduced through the combination of point-wise loss, gradient loss, surface normal loss, and the structural similarity index measure (SSIM) (Wang, Bovik, Sheikh, & Simoncelli, 2004) loss functions. The designed loss function learns to estimate the depth while minimizing the boundaries of scenes as well as the 3D structure of the faces. Fig. 7 shows an overview of the proposed loss function. The hybrid loss function L between g and d is defined as the weighted sum of the four different losses

$$L(g, d) = w_1 L_{depth}(g, d) + w_2 L_{SSIM}(g, d) + w_3 L_{grad}(g, d) + w_4 L_{SurfaceNorm}(g, d) \quad (1)$$

The first loss term (L_{depth}) represents the point-wise ($L1$) loss for the depth values and is according to Eq. (2).

$$L_{depth}(y, \check{y}) = \frac{1}{n} \sum_p |g_p - d_p| \quad (2)$$

The second loss term (L_{SSIM}) incorporates the SSIM metric with its upper bound for reconstructing the image using Eq. (3) (Wang et al., 2004).

$$L_{SSIM}(y, \check{y}) = \left(\frac{1 - L_{SSIM}(g, d)}{MaxDepth} \right) \quad (3)$$

The third term (L_{grad}) represents the ($L1$) loss for the gradient of the image depth with penalizing the error around their edges according to Eq. (4).

$$L_{grad}(g, d) = \frac{1}{n} \sum_p \nabla_x(e_p) + \nabla_y(e_p) \quad (4)$$

where $\nabla_x(e_p)$ and $\nabla_y(e_p)$ denote the spatial derivatives of the difference between the ground truth and predicted depth for the p^{th} pixels e_p which stands ($\|g_p - d_p\|$) for the x, y -axis. The depth maps gradient loss is sensitive to both x, y axes and is obtained using Sobel Filter method. It is important to note that the two loss functions presented, (L_{depth}) and (L_{grad}), complement each other for various types of errors. As a result, we use the (weighted) sum of (L_{depth}) and (L_{grad}).

According to the statistics of natural range images, depth maps of natural scenes can be roughly approximated by a limited number of smooth surfaces and step edges in between them. For example, at an object edge, depth is frequently discontinuous. Errors along such sharp edges are penalized by (L_{grad}). However, while depth differences at such occluding boundaries of objects might be very high, we must choose a reasonable value. We explore yet another loss to deal with such small depth structures and enhance fine details of depth maps. This loss measures the accuracy of the normal to the surface of an estimated depth map with respect to its ground truth.

The ($L_{SurfaceNorm}$) loss function is used to avoid the small structural errors and estimate the normal and predicted depth maps. The surface norms of the ground-truth and the predicted depth are denoted by

$$n_p^g = (\psi[-\nabla_x(g_p), -\nabla_y(g_p), 1]^T)$$

and

$$n_p^d = (\psi[-\nabla_x(d_p), -\nabla_y(d_p), 1]^T)$$

where n_p^g, n_p^d are the surface normal vectors, ∇ is a vector differential operator, ψ calculates the gradients of the difference between the ground truth and predicted depth in both the horizontal and vertical directions. The loss is computed by the difference between the two surfaces normal according to Eq. (5).

$$L_{SurfaceNorm} = \frac{1}{n} \sum_p \left(1 - \frac{\langle n_p^d, n_p^g \rangle}{\|n_p^d\| \cdot \|n_p^g\|} \right) \quad (5)$$

where $\langle n_p^d, n_p^g \rangle$ denotes the inner product of the vectors.

We empirically found and set the values of the weights w_1, w_2, w_3, w_4 as 0.28, 0.22, 0.30, 0.20 respectively. The four loss functions are evaluated through an adoptive method with varying weights and are coupled into a hybrid loss function for obtaining optimal results, the development procedure of our hybrid loss function is shown in Fig. 7.

6. Experiments

The experimental results are presented in this section to illustrate the effectiveness of the proposed method. We will start by comparing training and evaluation results of SoA to the proposed work and demonstrating a brief comparison analysis. Following that, the network was tested on four different test datasets. For the encoder, various comparison analyses have been conducted, analyzing them based on accuracy and computational footprints. Finally, we present an ablation study of the hybrid loss function, which will be used to demonstrate the benefits of the method. The proposed synthetic dataset was used to train all networks, which were then tested against different test datasets.

Our extensive experiments, which cover approximately four GPU months of computation, show that a model trained on a rich and diverse set of images, combined with an appropriate training procedure, yields SoA results in a variety of scenarios. To show this, zero-shot cross-dataset transfer protocol is used for comparison purposes. More specifically, the model was trained on one dataset and then evaluated on unseen test datasets.

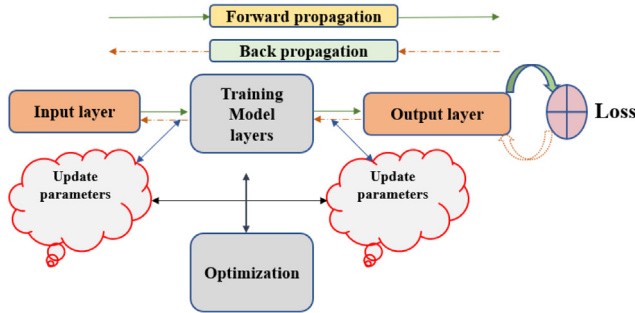


Fig. 8. Overall implementation details of training the proposed model with hybrid loss function.

6.1. Implementation details

The dataset was split into 0.8 and 0.2 ratios for training and validation, and the model was validated on four publicly available benchmark datasets (discussed in Section 6.2). The facial depth estimation model is trained using the PyTorch deep learning framework (Paszke et al., 2019). For all of the experiments, we use the Adam optimizer on a workstation equipped with NVIDIA 2080ti GPUs for 50 epochs with a 0.0001 learning rate and batch size of 6. For the entire model, there are approximately 14.42 million trainable parameters. For evaluations, Root Mean Square Error (RMSE), log Root Mean Square Error (RMSE (log)), Absolute Relative difference (AbsRel), Square Relative error (SqRel) and Accuracies are used, see Eqs. (6)–(10).

For training BTS (Lee et al., 2019), Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ is used and 10^{-6} learning is scheduled via polynomial decay from base learning rate 10^{-3} with power $p = 0.98$. The total number of epochs is set to 50 with batch size 4. The complete implementation details of the proposed model are illustrated in Fig. 8.

6.2. Test datasets

To benchmark the generalization performance of DESI networks (Alhashim & Wonka, 2018; Khan et al., 2021; Lee et al., 2019) and the proposed model trained on the synthetic human facial dataset with various pre-trained models such as (EfficientNet-B0, EfficientNet-B7, ResNet-101, ResNet-50, DenseNet-169, DenseNet-201, DenseNet-161), four datasets are selected based on diversity and accuracy of their ground truth. This includes Pandora (Borghi et al., 2017), Eurecom Kinect Face (Min et al., 2014), Biwi Kinect Head Pose (Fanelli et al., 2011) and our proposed test dataset for the testing and evaluation purposes. It should be noted rather than fine-tuning the networks, we have trained all the models from scratch on these datasets. We refer to this experimental procedure as zero-shot cross-dataset validation.

- **Pandora (Borghi et al., 2017):** Pandora dataset is used for different applications such as head pose estimation, head center localization, depth estimation and shoulder pose estimation. It contains a total of 250K full resolution RGB images with corresponding depth images.
- **Eurecom Kinect Face (Min et al., 2014):** The dataset consists of the multi-model face images of 52 people including 38 males and 14 females, which is obtained by using the Kinect sensor. It consists of different facial expression, occlusion and lighting conditions in 9 different states such as smile, eye occlusion, mouth, light and paper, neutral, open mouth, left–right profile.

- **Biwi Kinect Head Pose (Fanelli et al., 2011):** Consists of 15k images of 20 subjects recorded by using the Kinect sensor by moving the heads freely around each side. For every frame, RGB and depth images are provided, together with the 3D location of the head and its rotation angles.

6.3. Evaluation metrics

To evaluate the results a commonly accepted evaluation method has been used with five evaluation indicators: Root Mean Square Error (RMSE), log Root Mean Square Error (RMSE (log)), Absolute Relative difference (AbsRel), Square Relative error (SqRel), Accuracies, Normalized Root Mean Square Error (NRMSE) and R-squared. These are formulated as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i \in N} \|d_i - g_i\|^2} \quad (6)$$

$$RMSE_{Log} = \frac{1}{N} \sum_{i \in N} \|\log(d_i) - \log(g_i)\|^2 \quad (7)$$

$$AbsRel = \frac{1}{N} \sum_{i \in N} \frac{\|d_i - g_i\|}{g_i} \quad (8)$$

$$SqRel = \frac{1}{N} \sum_{i \in N} \frac{\|d_i - g_i\|^2}{g_i} \quad (9)$$

$$Accuracies = \% \text{ of } d_i \max\left(\frac{d_i}{g_i}, \frac{g_i}{d_i}\right) = \delta < thr \quad (10)$$

$$NRMSE = \frac{RMSE - RMSE_{min}}{RMSE_{max} - RMSE_{min}} \quad (11)$$

$$R^2 = 1 - \frac{\sum_{m=1}^N (d_i - g_i)^2}{\sum_{i=1}^N (d_i - \bar{g}_i)^2} \quad (12)$$

where g_i is the ground truth, \bar{g}_i is the mean of the ground truth and d_i is the predicted depth of the pixel i , N denotes the total number of pixels and thr denotes the threshold for determining the accuracy.

6.4. Comparison of encoders

Since the proposed network uses existing models as an encoder for dense feature extraction, it is worth comparing its output to that of other commonly used base networks for similar tasks. We checked the proposed method by adjusting the encoder with different models while keeping the other settings the same. The influence of the encoder architecture is illustrated in Fig. 10. The model is trained with EfficientNet-B0, EfficientNet-B7, ResNet-101, ResNet-50, DenseNet-169, DenseNet-201, DenseNet-161 encoder as our baseline architectures and the relative improvement in performance when swapping with different encoders. The results are reported in Table 1 (row 2,3, 5–9).

6.5. Final results and comparison with prior work

Results achieved with the proposed methodology are summarized in Fig. 9 and Table 1, the performance of the facial depth estimation model is compared to the SoA on the synthetic human facial dataset. As it can be seen from Table 1, the proposed network achieves SoA results.

Table 1

Comparison of various depth estimation models with the proposed method FaceDepth, BTS (Lee et al., 2019), Densedept (Alhashim & Wonka, 2018) and UNet-simple (Khan et al., 2021) with various base models (EfficientNet-B0, EfficientNet-B7, ResNet-101, ResNet-50, DenseNet-201, DenseNet-161). FC refers to the facial crop which means the errors are estimated only on the facial region.

No.	Methods	AbsRel	SqRel	RMSE	NRMSE	R^2	RMSElog	$\delta_1 < 1.25$	$\delta_2 < 1.25^2$	$\delta_3 < 1.25^3$
1.	DenseDepth-161	0.0312	0.0121	0.0610	0.0607	0.0345	0.0169	0.9854	0.9876	0.9902
2.	DenseDepth-121	0.0320	0.0132	0.0712	0.0746	0.0465	0.0180	0.9732	0.9803	0.9880
3.	DenseDepth-169	0.0296	0.0096	0.0373	0.0432	0.0245	0.0129	0.9890	0.9920	0.9981
4.	BTS	0.0165	0.0092	0.0206	0.0321	0.0254	0.0102	0.9830	0.9943	0.9956
5.	DenseDepth-201	0.0375	0.0097	0.0304	0.0476	0.0265	0.0101	0.9920	0.9956	0.9969
6.	ResNet-101	0.0123	0.0210	0.0306	0.0456	0.0236	0.0089	0.9938	0.9965	0.9980
7.	ResNet-50	0.0232	0.0219	0.0445	0.0598	0.0231	0.0186	0.9919	0.9974	0.9984
8.	EfficientNet-B0	0.0145	0.0280	0.0360	0.0476	0.0228	0.0154	0.9912	0.9934	0.9978
9.	EfficientNet-B7	0.0132	0.0234	0.0353	0.0431	0.0225	0.0144	0.9880	0.9909	0.9965
10.	UNet-simple	0.0103	0.0207	0.0281	0.0321	0.0212	0.0089	0.9960	0.9976	0.9987
11.	UNet-simple (FC)	0.0098	0.0096	0.0143	0.0274	0.0201	0.0043	0.9982	0.9992	0.9996
12.	DenseDepth(FC)-169	0.0110	0.0074	0.0161	0.0286	0.0189	0.0034	0.9981	0.9990	0.9992
13.	BTS(FC)	0.0109	0.0072	0.0152	0.0248	0.0165	0.0033	0.9971	0.9991	0.9992
14.	ResNet (FC)-101	0.0132	0.0077	0.0170	0.0213	0.0149	0.0035	0.9980	0.9990	0.9992
15.	EfficientNet (FC)-B7	0.0112	0.0076	0.0166	0.0210	0.0141	0.0032	0.9887	0.9945	0.9989
16.	Our FaceDepth (FC)	0.0176	0.0030	0.0105	0.0204	0.0136	0.0029	0.9982	0.9986	0.9996

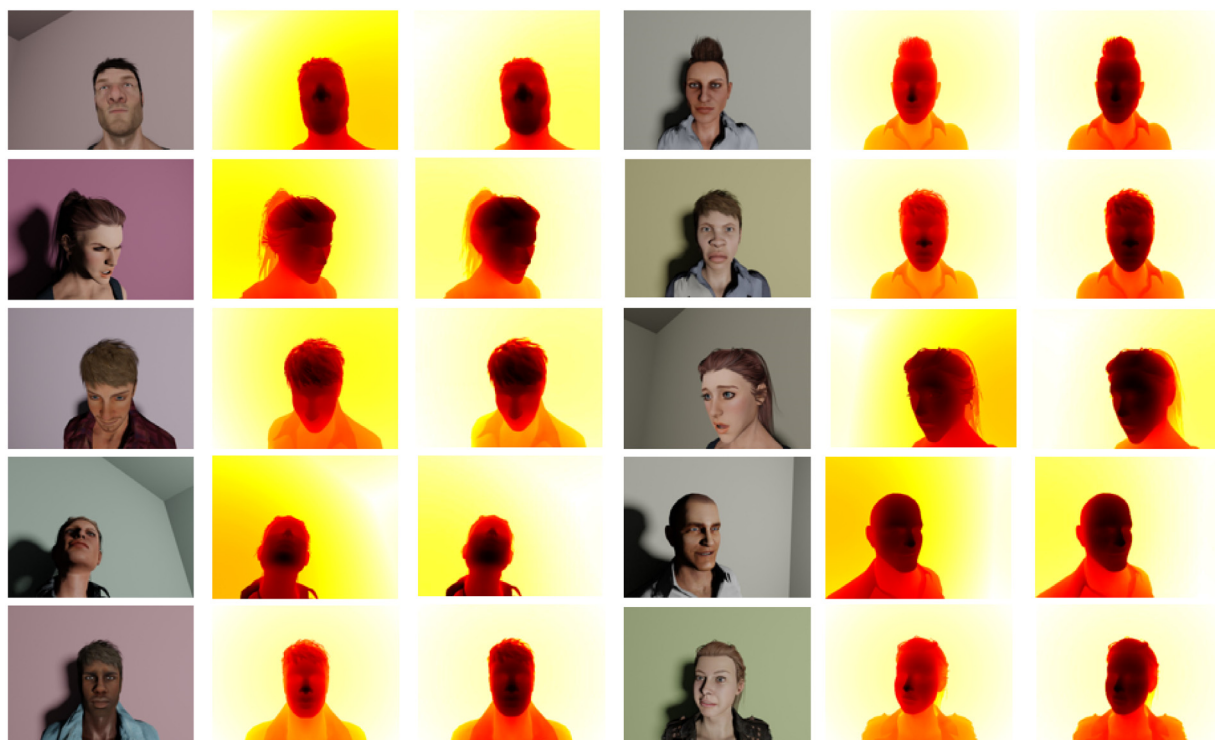


Fig. 9. Qualitative results of the proposed method on a subset of the synthetic human facial dataset that was not used for training or validation. From left to right, input RGB images, ground truth depth images and predicted depth images.

As stated in Section 4, since there are no available benchmark methods for performance evaluation; in the first phase the generated synthetic human facial dataset is utilized to retrain the SoA DESI methods (Alhashim & Wonka, 2018; Lee et al., 2019) and a UNet-simple (Khan et al., 2021). Afterwards, all the trained models are then evaluated and tested on four benchmark datasets. As stated above, the model is initially trained over the whole image and then applied to the Facial crop (FC) for evaluating errors particularly in the face region. In other words, the depth has been masked within a certain range of 50 centimeters from the camera to evaluate the results only on the facial region of the images, see Table 1 (rows 11–16). The proposed lightweight network structure contains fewer parameters to the SoA methods. A detailed comparison analysis is given in Table 2.

6.6. Qualitative result

We discuss qualitative results from the proposed framework against SoA methods in this section. Figs. 10 and 11 show a qualitative comparison of our model to the three best-performing models with various Encoders architectures. As it can be observed from Fig. 10 our results show better information and consistency, which proves that the proposed method performs better at depth estimation with improvements on the facial region.

In testing across a combination of real and synthetic images, we outperform SoA both quantitatively and qualitatively, and set a new SoA for Facial DESI. Example results are shown in Table 1, Table 2 and Fig. 11.

In terms of accuracy and depth range, based on the evaluations the proposed method achieved the best performance as compared to other SoA methods. On the synthetic human facial dataset,

Table 2

Properties of the studied methods (Lee et al., 2019), (Alhashim & Wonka, 2018), UNet-simple (Khan et al., 2021) and our proposed model (ED: Encoder–Decoder; F: Trained on the synthetic human facial dataset); LR/E: Learning Rate/Epochs; CC: Computational Complexity.

Method	Input	Type	Optimizer	Parameters	Output	LR/E	CC
BTS	640 × 480F	ED	Adam	46.6M	640 × 480F	0.0001/50	69.23 GMac
DenseDepth-169	640 × 480F	ED	Adam	42.6M	320 × 240F	0.0001/20	66.12 GMac
ResNet-50	640 × 480F	ED	Adam	68M	640 × 480F	0.0001/25	101.27 GMac
EfficientNet-B7	640 × 480F	ED	Adam	80.4M	640 × 480F	0.00001/20	113.44 GMac
UNet-simple (FC)	640 × 480F	UNet	Adam	17.27M	640 × 480F	0.001/20	188.04 GMac
Our FaceDepth	640 × 480F	ED	Adam	14.42M	320 × 240F	0.0001/50	16.41 GMac

Table 3

Experimental results using a synthetic human facial dataset with various weights setting.

Method	w_1, w_2, w_3, w_4	AbsRel	SqRel	RMSE	RMSElog	$\delta_1 < 1.25$	$\delta_2 < 1.25^2$	$\delta_3 < 1.25^3$
FaceDepth [FC]	1.00, 0.1, 0.1, 1.00	0.0118	0.0037	0.0108	0.0031	0.9982	0.9985	0.9996
FaceDepth [FC]	1.00, 0.00, 0.00, 0.00	0.0178	0.0048	0.0124	0.0042	0.9961	0.9974	0.9991
FaceDepth [FC]	0.00, 1.00, 0.00, 0.00	0.0107	0.0011	0.0108	0.0033	0.9888	0.9924	0.9945
FaceDepth [FC]	0.00, 0.00, 1.00, 0.00	0.0495	0.0086	0.0181	0.0081	0.9881	0.9952	0.9986
FaceDepth [FC]	0.00, 0.00, 0.00, 1.00	0.0039	0.0206	0.0256	0.0113	0.8781	0.9821	0.9840
FaceDepth [FC]	0.25, 0.25, 0.25, 0.25	0.0219	0.0038	0.0109	0.0032	0.9961	0.9982	0.9990
FaceDepth [FC]	0.28, 0.22, 0.30, 0.20	0.0176	0.0030	0.0105	0.0029	0.9982	0.9986	0.9996

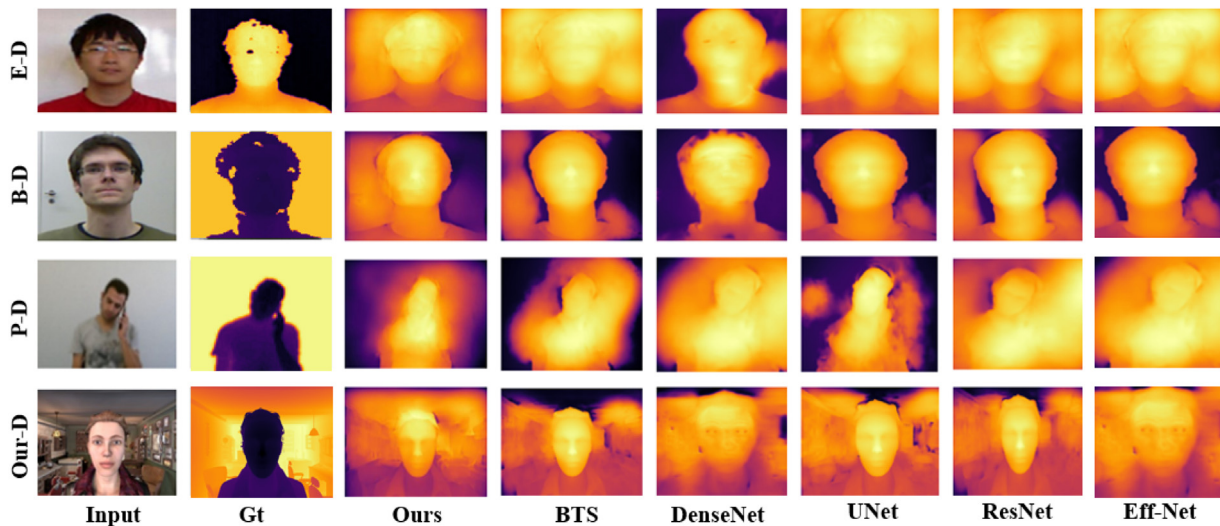


Fig. 10. A qualitative comparison of our approach to the four best competitors: from left to right; (Input: input RGB images; GT: ground truth images; Ours: Our FaceDepth method; BTS (Lee et al., 2019), Ef-Net: EfficientNet-B7 (Alhashim & Wonka, 2018; Wang et al., 2019); Rs-Net: ResNet-50 (Alhashim & Wonka, 2018; He, Zhang, Ren, & Sun, 2016); D-Net: DenseDepth-169 (Alhashim & Wonka, 2018); U-Net: UNet-simple (FC) (Khan et al., 2021) applied to different datasets (Our-D: Synthetic human facial dataset; P-D: Pandora dataset (Borghi et al., 2017); E-D: Eurecom Kinect Face dataset (Min et al., 2014); B-D: Biwi Kinect Head Pose dataset (Fanelli et al., 2011).

the proposed network achieved 0.0105 RMSE and threshold accuracy of 0.9996 with $\delta < 1.25^3$ as shown in Table 1 (row 16). Furthermore, the proposed method is shown to have a significantly reduced memory footprint with improved computational efficiency as compared to other SoA methods as shown in Table 2 (row 6). At 16.41 G-MACs per frame, this approach can enable real time single frame depth estimation. Table 2 (row 5) portrays that albeit the UNet-Simple model has comparatively lower number of parameters comparing to the other models; however, the design principal of double convolution layer, where the batch norm, ReLU activation and the bi-linear up-sampling stages make it computationally expensive. Moreover, our faceDepth model has a fewer parameters with pre-trained weights help in avoiding several computational steps in the decoder and thereby reducing the computational complexity.

Table 2 shows properties of the studied methods for single image facial depth estimation (ED: Encoder–Decoder; F: Trained on the synthetic human facial dataset). Based on our evaluations, BTS (Lee et al., 2019), DenseDepth (Alhashim & Wonka, 2018) with various base models and UNet-simple method (Khan et al.,

2021) can generate high resolution depth maps with comparable accuracy but they are computationally expensive and require a significant amount of memory. On the other hand, FaceDepth significantly reduced the computational time and memory footprint, which can be used for both quality and low-cost single frame facial depth estimations (Table 2 and Fig. 11).

6.7. Ablation study

The ablation studies in Table 3 are performed adaptively such that all the possibilities of coupling the terms in connection with their corresponding weights are tested and their performance is recorded and thereby based on the optimal predicted depth output the four terms combination has been selected.

We conduct ablation studies to analyze the effectiveness of the hybrid loss criteria utilized in the proposed network architecture. We start with weights defined for loss function in Eq. (1). The result is given in Table 3. As the total weights ($w_1 = 0.28, w_2 = 0.22, w_3 = 0.30, w_4 = 0.20$) sum is equal to 1, the overall performance is improved. We also analyze the effect of weights separately and the results are shown in Table 3.

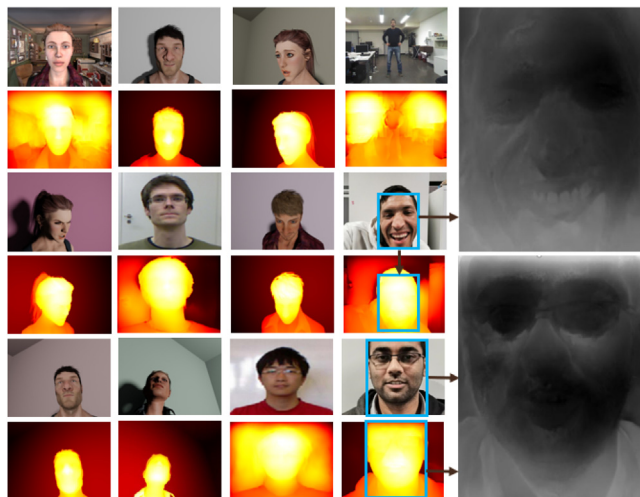


Fig. 11. Results of the baseline model trained using our proposed hybrid loss function and synthetic human facial dataset. The model trained using the hybrid loss function provides more details of local depth structure and higher accuracy at depth boundaries. The test images are a combination of real and synthetic images which is not used in the training process for any of the above models. Best viewed zoomed in on-screen shown on two real images.

As an exhaustive search of possible weight values is not computationally feasible, this study sought to show that no single element of the loss function can provide the demonstrated accuracy without the other methods.

This was done by setting the weights to 0 for all methods except the one being examined and is shown in rows 2–5 of Table 3 these rows should be compared with row 6 where each weight was set to the same value summing to 1 (0.25, 0.25, 0.25, 0.25). The best weight set examined is in row 7 (0.28, 0.22, 0.30, 0.20) which seems to indicate the relative importance L1 loss, particularly L1 calculated over the image gradient so as to magnify the significance of errors on edges.

One unexpected result is shown in row 5 where w_4 was set to 1 while all other weights were set to 0. This is the best result on the AbsRel metric but performs poorly on the rest.

One possibility is that if w_4 is too high, the network can prioritize the reduction of differences that are due to noise, and focus too much on the reduction small structural errors at the possible expense of errors around edges. This is supported by the fact that our best performing experiment in Table 3 had the lowest non zero value for w_4 .

It is a reasonable expectation that when only the surface norm is used in loss calculations that this would have the greatest impact on the relative absolute error but it is unclear why this did not translate into a greater improvement for AbsRel in the case that L1 was used for training as the primary difference between the loss function used in training and the evaluation metric is the scaling (g_i) factor. A more thorough ablation study analyzing this possibility may be investigated in future work.

7. Discussion

This research offers a new encoder–decoder model for facial depth estimation using synthetic human facial dataset and evaluates its performance against other SoA approaches. In contrast to the different SoA approaches, the developed framework has a remarkably smaller network size and reduced computational complexity. The performance significance is due to the model training method, which selects an adequately appropriate loss function through a combination of different loss functions and

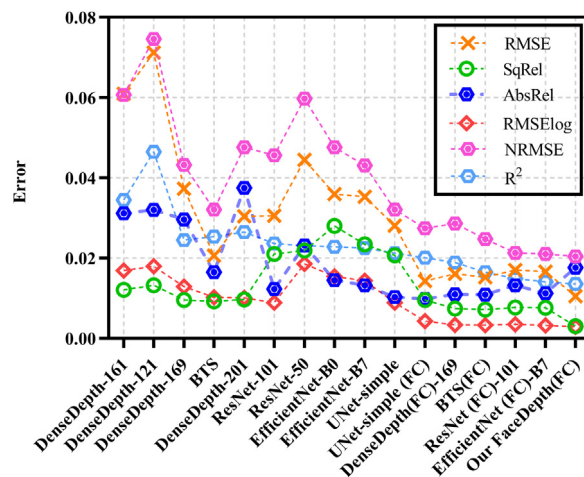


Fig. 12. The relative performance of several technique evaluation metrics (lower is better).

the use of a synthetic human facial dataset with pixel-accurate ground truth depth information.

The generated synthetic human facial depth dataset is analyzed using a set of SoA DESI neural networks. This work utilized DESI neural networks to train over the generated synthetic dataset and evaluate with test data because there are no publicly available benchmarks techniques for evaluations. A new CNN model is also proposed, and its performance is compared to the SoA methods. The performances of the proposed model and the SoA methods were measured using seven evaluation matrices: Root Mean Square Error (RMSE), log Root Mean Square Error (RMSE (log)), Absolute Relative Difference (AbsRel), Square Relative Error (SqRel), Accuracies, Normalized Root Mean Square Error (NRMSE), and R-squared shown in Table 1. In addition, when compared to previous SoA approaches, the suggested method has a much smaller memory footprint and improved computational efficiency, as demonstrated in Table 2 (row 6). At 16.41 G-MACs per frame, this approach can enable real time single frame depth estimation.

We test on a collection of datasets that were never seen during training for all the experiments and comparisons to the SoA. Figs. 10 and 11 illustrate a qualitative comparison of the models, which show that the proposed method performs better at depth estimation generalization with improvements in the facial region. Following that, we adaptively run ablation tests on the loss function Table 3, in which all possible couplings of terms with their corresponding weights are examined and their performance is recorded, and the four terms combination is chosen based on the optimal predicted depth output. A comparison of the different types of error concerning the SoA approaches is illustrated in Fig. 12. It is evident high-performance achievement with the proposed method by reducing the errors across many test datasets compared to the different SoA approaches. The selection of appropriate loss function and the synthetic dataset enables the model to reduce the error with lower computational cost. The model performance in reducing the different types of errors is shown through a box plot in Fig. 13. In general, the proposed model reduces all the errors, while particularly, it has a significant performance for the error types RMSElog and SqRel compared to the AbsRel and RMSE, respectively.

Synthetic data can have a lot of advantages. Ground truth is perfect and available for tasks such as depth estimation, head pose, reconstruction, tracking, and camera or object position without the need for costly human labeling. Motion blur and

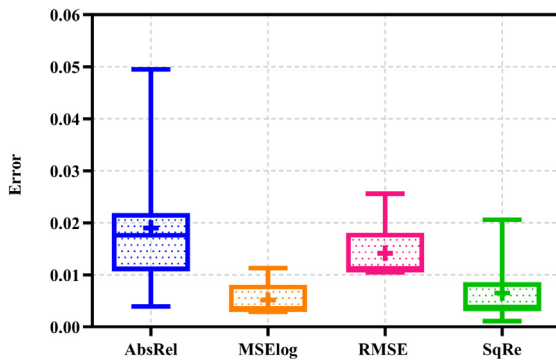


Fig. 13. The FaceDepth method box plot shows the relative performance of various errors.

lighting changes, as well as camera position and expressions for algorithm introspection, can all be used to recreate sequences. It is also possible to generate conditions that would be impossible to replicate in real life, such as exact ground truth depth information. We would need a large number of images dataset containing pixel-accurate ground truth of a scene to train and test deep learning algorithms making it suitable for deployment in embedded systems and in Edge-AI application. Many other related challenges, such as shape completion, 3D reconstruction, and 3D fusion may make use of synthetic data necessary for the real-life applications.

8. Conclusion

The principle contribution of this research is an improved and efficient encoder–decoder based neural model for single image frame depth estimation. This model is competitive with other SoA depth estimation models, but is significantly smaller in size and computational complexity, making it suitable for deployment in embedded systems and in Edge-AI applications (Ignatov et al., 2018).

When tested across four public data sets, this model shows performance that is equal to or better than SoA across all primary metrics, as shown in Section 6.2 and Table 1. In part this level of performance relies on a training methodology, which makes use of synthetic data samples to provide a pixel-accurate ground truth for depth. This improves on ground truth data available from existing public datasets, and is a major contributory factor to the high performance and lower complexity of the model. A second significant contribution of this work is the synthetic training dataset and associated training methodology which are described in detail in this work.

A key take-away from this research is that synthetic human facial data can provide higher quality ground truth depth data than can be obtained in practical data acquisition and this high-quality training data can be leveraged to achieve improved, lightweight, single image depth models. Further improvement beyond SoA should be feasible by introducing real-data samples, improving the photo-realism of the synthetic data samples and introducing a wider variety of facial features, expressions and scene lightings.

Thus future work could include investigations into the super-positioning of photo-realistic face textures over the synthetic avatar models and introducing more sophisticated facial dynamics such as mouth and eye variations used to express a wide range of emotions. Also of interest would be an exploration of different lightweight encoder–decoder architectures, data augmentation techniques, and evaluations with a broader range of test datasets. It would also be interesting to explore some 3D loss functions to address specific downstream applications.

Finally, the release of the synthetic human facial depth dataset used in this research and the associated 3D synthetic subject models, will benefit future research in areas such as 3D facial reconstruction, understanding, and facial analysis.

CRediT authorship contribution statement

Faisal Khan: Formal analysis, Investigation, Methodology and first draft, Data preparation, Writing – original draft, Conceptualization, Software, Training and evaluation. **Shahid Hussain:** Technical guideline in system modeling, Synthetic dataset generation process, Composition of hybrid loss function, Flowchart, Overall draft preparation. **Shubhajit Basak:** Data creation, Proposed the hybrid loss function. **Joseph Lemley:** Review and editing the draft, Evaluating the hybrid loss function, Explanation of the combined loss behavior. **Peter Corcoran:** Supervision, Validation, Project administration, Final draft preparation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Abdelmounaime, S., & Dong-Chen, H. (2013). New brodatz-based image databases for grayscale color and multiband texture analysis. In *International Scholarly Research Notices*, Vol. 2013. Hindawi.
- Alhashim, I., & Wonka, P. (2018). High quality monocular depth estimation via transfer learning. ArXiv Preprint arXiv:1812.11941.
- Andraghetti, L., Myriokefalitakis, P., Dovesi, P. L., Luque, B., Poggi, M., Pieropan, A., et al. (2019). Enhancing self-supervised monocular depth estimation with traditional visual odometry. In *2019 International Conference on 3D Vision (3DV)* (pp. 424–433). IEEE.
- Athira, M. V., & Khan, D. M. (2020). Recent trends on object detection and image classification: A review. In *2020 International Conference on Computational Performance Evaluation (ComPE)* (pp. 427–435). <http://dx.doi.org/10.1109/ComPE49325.2020.9200080>.
- Basha, T., Avidan, S., Hornung, A., & Matusik, W. (2012). Structure and motion from scene registration. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1426–1433).
- Bazrafkan, S., Hossein, J., Joseph, L., & Corcoran, P. (2017). Semiparallel deep neural network hybrid architecture: first application on depth from monocular camera. *Journal of Electronic Imaging*, 4, 043–041.
- Bhat, S. F., Alhashim, I., & Wonka, P. (2020). Adabins: Depth estimation using adaptive bins. ArXiv Preprint arXiv:2011.14141.
- Borghini, G., Venturini, M., Vezzani, R., & Cucchiara, R. (2017). Poseidon: Face-from-depth for driver pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4661–4670).
- Chang, J., & Wetzstein, G. (2019). Deep Optics for Monocular Depth Estimation and 3D Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Chen, Y., Zhao, H., Hu, Z., & Peng, J. (2021). Attention-based context aggregation network for monocular depth estimation. *International Journal of Machine Learning and Cybernetics*, 1–14.
- Choi, H., Lee, H., Kim, S., Kim, S., Kim, S., & Min, D. (2020). Adaptive confidence thresholding for semi-supervised monocular depth estimation. ArXiv Preprint arXiv:2009.12840.
- Elanattil, S., & Moghadam, P. (2019). Synthetic human model dataset for skeleton driven non-rigid motion tracking and 3D reconstruction. ArXiv Preprint arXiv:1903.02679.
- Fan, D.-P., Li, T., Lin, Z., Ji, G.-P., Zhang, D., Cheng, M.-M., et al. (2021). Re-thinking co-salient object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Fanelli, G., Weise, T., Gall, J., & Van Gool, L. (2011). Real time head pose estimation from consumer depth cameras. In *Joint Pattern Recognition Symposium* (pp. 101–110).
- Fu, H., Gong, M., Wang, C., Batmanghelich, K., & Tao, D. (2018). Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2002–2011).
- Goldman, M., Hassner, T., & Avidan, S. Learn stereo, infer mono: Siamese networks for self-supervised, monocular, depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

- Gu, J., Yang, X., De Mello, S., & Kautz, J. (2017). Dynamic facial analysis: From bayesian filtering to recurrent neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1548–1557).
- Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., & Gaidon, A. (2020). 3D Packing for Self-Supervised Monocular Depth Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).
- Ignatov, A., Timofte, R., Chou, W., Wang, K., Wu, M., Hartley, T., et al. (2018). AI Benchmark: Running Deep Neural Networks on Android Smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*.
- Javidnia, H., & Corcoran, P. (2017). Accurate depth map estimation from small motions. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 2453–2461).
- Jiang, J., El-Shazly, E. H., & Zhang, X. (2019). Gaussian weighted deep modeling for improved depth estimation in monocular images. *IEEE Access*, 7, 134718–134729.
- Johnston, A., & Carneiro, G. (2020). Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4756–4765).
- Khan, F., Basak, S., & Corcoran, P. (2021). Accurate 2D facial depth models derived from a 3D synthetic dataset. In *2021 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1–6). <http://dx.doi.org/10.1109/ICCE50685.2021.9427595>.
- Khan, F., Salahuddin, S., & Javidnia, H. (2020). Deep learning-based monocular depth estimation methods—A state-of-the-art review. *Sensors*, 20(8), 2272.
- Klingner, M., Termöhlen, J.-A., Mikolajczyk, J., & Fingscheidt, T. (2020). Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *European Conference on Computer Vision* (pp. 582–600). Springer.
- Koo, H.-S., & Lam, K.-M. (2008). Recovering the 3D shape and poses of face images based on the similarity transform. *Pattern Recognition Letters*, 29(6), 712–723.
- Kuznietsov, Y., Proesmans, M., & Van Gool, L. CoMoDA: Continuous Monocular Depth Adaptation Using Past Experiences. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (pp. 2907–2917).
- Laidlow, T., Czarnowski, J., & Leutenegger, S. (2019). Deepfusion: real-time dense 3D reconstruction for monocular SLAM using single-view depth and gradient predictions. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 4068–4074).
- Lee, J. H., Han, M.-K., Ko, D. W., & Suh, I. H. (2019). From big to small: Multi-scale local planar guidance for monocular depth estimation. ArXiv Preprint [arXiv:1907.10326](https://arxiv.org/abs/1907.10326).
- Lee, J.-H., & Kim, C.-S. (2020). Multi-loss rebalancing algorithm for monocular depth estimation. In *Proceedings of the 2020 European Conference on Computer Vision (ECCV)*, Glasgow, UK (pp. 23–28).
- Lei, Z., Wang, Y., Li, Z., & Yang, J. (2021). Attention based multilayer feature fusion convolutional neural network for unsupervised monocular depth estimation. *Neurocomputing*, 423, 343–352.
- Li, R., He, X., Xue, D., Su, S., Mao, Q., Zhu, Y., et al. (2021). Learning depth via leveraging semantics: Self-supervised monocular depth estimation with both implicit and explicit semantic guidance. ArXiv Preprint [arXiv:2102.06685](https://arxiv.org/abs/2102.06685).
- Liu, P., Zhang, Z., Meng, Z., & Gao, N. (2020). Joint attention mechanisms for monocular depth estimation with multi-scale convolutions and adaptive weight adjustment. *IEEE Access*, 8, 184437–184450.
- Min, R., Kose, N., & Dugelay, J.-L. (2014). Kinectfacedb: A kinect database for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(11), 1534–1548.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.
- Poggi, M., Aleotti, F., Tosi, F., & Mattocchia, S. On the uncertainty of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 3227–3237).
- Ranftl, R., Bochkovskiy, A., & Koltun, V. (2021). Vision transformers for dense prediction. ArXiv Preprint [arXiv:2103.13413](https://arxiv.org/abs/2103.13413).
- Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., & Koltun, V. (2020). Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.
- Roy-Chowdhury, A. K., & Chellappa, R. (2005). Statistical bias in 3-D reconstruction from a monocular video. *IEEE Transactions on Image Processing*, 14(8), 1057–1062.
- dos Santos Rosa, N., Guizilini, V., & Grassi, V. (2019). Sparse-to-continuous: Enhancing monocular depth estimation using occupancy maps. In *2019 19th International Conference on Advanced Robotics (ICAR)* (pp. 793–800). IEEE.
- Schöps, T., Sattler, T., Häne, C., & Pollefeys, M. (2017). Large-scale outdoor 3D reconstruction on a mobile device. *Computer Vision and Image Understanding*, 157, 151–166.
- Sifre, L., & Mallat, S. (2014). Rigid-motion scattering for texture classification. *Applied and Computational Harmonic Analysis*, 00, 01–20.
- Song, X., Li, W., Zhou, D., Dai, Y., Fang, J., Li, H., et al. (2021). MLDA-net: Multi-level dual attention based network for self-supervised monocular depth estimation. *IEEE Transactions on Image Processing*.
- Song, M., Lim, S., & Kim, W. (2021). Monocular depth estimation using Laplacian pyramid-based depth residuals. *IEEE Transactions on Circuits and Systems for Video Technology*.
- Spencer, J., Bowden, R., & Hadfield, S. (2020). DeFeat-Net: General Monocular Depth via Simultaneous Unsupervised Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 14402–14413).
- Tian, Y., & Hu, X. (2021). Monocular depth estimation based on a single image: a literature review. 11720, In *Twelfth International Conference on Graphics and Image Processing (ICGIP 2020)* (p. 117201Z). International Society for Optics and Photonics.
- Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., et al. (2017). Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 109–117).
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612.
- Wang, H., Yang, J., Liang, W., & Tong, X. (2019). Deep single-view 3d object reconstruction with visual hull embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, (pp. 8941–8948).
- Wang, Z., Yu, Z., Zhao, C., Zhu, X., Qin, Y., Zhou, Q., et al. (2020). Deep spatial gradient and temporal depth learning for face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5042–5051).
- Ware, C. (2019). *Information Visualization: Perception for Design*. Morgan Kaufmann.
- Wenxian, H. (2010). *A Study of Fast, Robust Stereo-Matching Algorithms*. Cambridge, Massachusetts: MIT.
- Widya, A. R., Monno, Y., Okutomi, M., Suzuki, S., Gotoda, T., & Miki, K. (2021). Self-supervised monocular depth estimation in gastroendoscopy using GAN-augmented images. In *Medical Imaging 2021: Image Processing*, Vol. 11596. International Society for Optics and Photonics, Article 1159616.
- Xian, K., Zhang, J., Wang, O., Mai, L., Lin, Z., & Cao, Z. (2020). Structure-guided ranking loss for single image depth prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 611–620).
- Ye, X., Chen, S., & Xu, R. (2021). Dpnet: Detail-preserving network for high quality monocular depth estimation. *Pattern Recognition*, 109, Article 107578.
- Yin, W., Liu, Y., Shen, C., & Yan, Y. (2019). Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 5684–5693).
- Yue, M., Fu, G., Wu, M., & Wang, H. (2020). Semi-supervised monocular depth estimation based on semantic supervision. *Journal of Intelligent and Robotic Systems*, 100, 455–463.
- Yusiong, J. P. T., & Naval, P. C. (2020). A semi-supervised approach to monocular depth estimation, depth refinement, and semantic segmentation of driving scenes using a siamese triple decoder architecture. *Informatica*, 44(4).
- Zhao, Y., Jin, F., Wang, M., & Wang, S. (2020). Knowledge graphs meet geometry for semi-supervised monocular depth estimation. In *International Conference on Knowledge Science, Engineering and Management* (pp. 40–52). Springer.

Chapter 4

Face Reconstruction with weak Supervision

4.1 Background

When it comes to facial analysis tasks, single image-based face reconstruction has received considerable attention in the computer vision community. Predicting the 3D features of the human face is the pre-requisite for many facial analysis tasks such as face reenactment and speech-driven animation, video dubbing, projection mapping, face replacement, facial animations, and many others [170]. Due to the limitation of depth sensors, it is difficult to capture high-frequency details through RGB-D data. At the same time learning from synthetic face depth is only able to predict the mean shape. Capturing high-quality 3D scans is expensive and often restricted because of ethical and privacy concerns. A popular alternative to these facial capturing methods is to estimate the face geometry from an uncalibrated 2D face image. However, this 3D-from-2D reconstruction of the human face is an ill-posed problem because of the complexity and variations of the human face. We need to capture the individual facial geometry, head pose, and texture information such as color and illumination. A common solution is to add some prior knowledge about the human face, as human faces have a common mean shape. One of the well-accepted methods to add this prior knowledge is starting with a statistical model of the human face.

One of the most popular of these models is called the 3D Morphable model (3DMM) [20]. 3DMMs are linear statistical models of shape and appearance that are built from a set of 3D scans that provides an analytical definition of the human face and acts as a priori to novel face synthesis tasks. With the help of 3DMMs, 3D face reconstruction can be formulated as a non-linear optimization problem that is constrained by linear statistical models of shape

and facial texture. With the advancement of deep learning, recent methods directly try to learn the mapping between the 2D image and 3D faces, encoding the prior knowledge in the weights of the learned models. But the main obstacle when applying deep learning to 3D face reconstruction is the lack of facial scans or high-quality depth as the ground truth data. A solution to this is to create synthetic 3D faces with the help of 3DMMs. But as 3DMMs are used to create the ground truth, it does not overcome the shortcomings of the linearly modeled data and fails to provide enough variations to the ground truths. To overcome this problem of collecting ground truth scans and to avoid the drawbacks of synthetic sets, a new strategy is introduced based on self-supervision. The main idea behind this approach is that the generated data by the learning network itself provides supervision by adding a rendering layer at the end of the network [113, 130]. The rendering function is fully differentiable, and the rendering parameters can be learned. This enables an end-to-end learnable network as follows -

1. A 3D face is synthesized by learning the latent parameters of shape and textures through a regressor network.
2. It is then rendered by a differentiable renderer with the help of camera and illumination parameters.
3. The reconstruction error between the rendered and ground truth face images is calculated by preferred matrices.
4. The parameters of the regressor and the differentiable renderer are updated based on the derivatives of the error.

4.2 Research Objective

The above-mentioned regressor network predicts the 3DMM parameters via a deep neural network. Almost all of the previous methods [113, 112, 54, 40, 58, 138] used CNN-based backbones like Resnet [67] to extract the 3DMM parameters. But by its fundamental design, convolution operations are local to the image space and sometimes incapable of processing global operations. Adding skip connections can overcome some of these shortcomings but sometimes fail to extract fine spatial information because of misalignment of the features extracted in different layers. In recent times, transformer networks have become popular in computer vision tasks because of their ability to capture long-term dependencies. Specifically, vision transformers (ViTs) [44] have achieved SOTA performance in different computer vision tasks like object detection [122], image segmentation [167], image classification [29],

etc. But as it extracts the long-term dependencies, sometimes it fails to learn the local features. So we need to incorporate the local feature learning abilities into the ViTs and see their performance in 3D face reconstruction tasks, which is not studied yet. As the pipeline does not use any ground truth face scans, the model performance highly relies on objective functions. So the effects of multi-loss functions are also studied to show the individual influence of the loss terms.

4.3 Summary of Contribution

The work is presented in the article - Basak, Shubhajit, Peter Corcoran, Rachel McDonnell, and Michael Schukat. "3D face-model reconstruction from a single image: A feature aggregation approach using hierarchical transformer with weak supervision." *Neural Networks* 156 (2022): 108-122. A copy of the paper is attached at the end of this chapter.

The contributions of the authors for the research mentioned above work [13] as per the four major criteria discussed in section 1.4 is presented in the table 4.1.

Table 4.1 Author's Contribution to [13]

Contribution Criteria	Contribution Percentage
Ideation	SB 100%
Experiments & Implementations	SB 100%
Manuscript Preparation	SB 80%, RM 5%, MS 5%, PC 10%
Background Work	SB 70%, MS 20%, PC 10%

To achieve the research objective, we have proposed to replace the regressor with a vision transformer. Normal ViTs are comparatively large and computationally expensive. Instead, we have used the Swin Transformer [95] as the backbone of the feature extractor. To add emphasis on the local feature extraction, we first introduced a hierarchical feature extractor consisting of four stages that extracted the features in four different resolutions. Then we gradually aggregate the features from the various stages through a multi-scale feature aggregation module (FAM), which fuses the multi-scale features and performs a coarse-to-fine feature extraction.

We have used the popular 300W-LP [168] database for this learning. While training the network, we utilized multi-loss function settings with some weak supervision. It consists of five different components, of which the first three are learned in an unsupervised manner - 1) Landmark Loss - which provides weak supervision with the help of a SOTA face alignment network [26]. 2) Photometric Loss - It measures the photometric discrepancies between the rendered image and the ground truth. 3) Perceptual Loss - a pretrained SOTA face

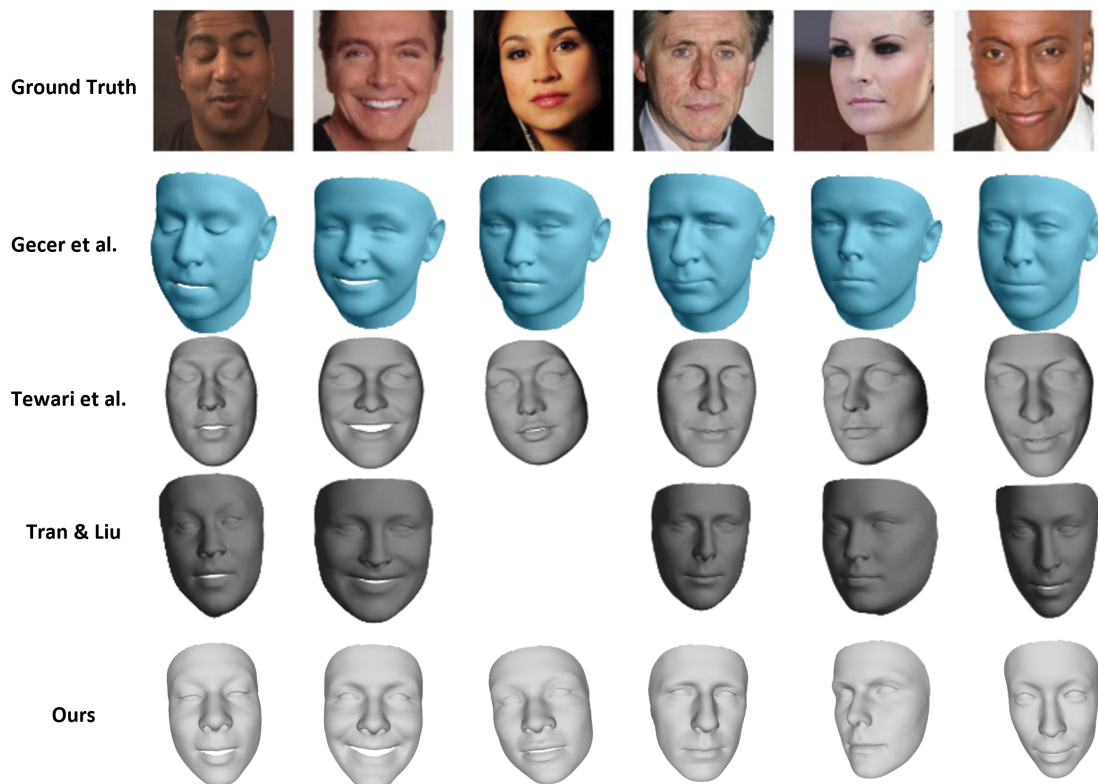


Fig. 4.1 Qualitative comparison of the generated face shape with previous works - GANFIT [57], Tewari et al. [129], Tran & Liu [138]. The results of the previous works are taken from GANFIT [57]

recognizer(FR) ArcFace [39] is used to calculate the perceptual features of the face and help to learn the more intrinsic characteristics of the face. The other two are supervised losses - 4) Shape Loss - we add a supervised cue by adding the shape loss, which is calculated as the L1 loss between the predicted 3DMM shape parameters and the ground truth 3DMM parameters. Adversarial Loss - To keep the predicted 3DMM parameters near to ground truth, we add an adversarial loss where a discriminator is trained to discriminate the fake shapes learned by the network from the real shape generated from the 300W-LP dataset. This helps generalize the network while balancing the unsupervised and supervised training.

4.4 Discussion on Contribution

This work provides the first-ever use of transformer networks in monocular face reconstruction tasks. Though there are multiple studies on face reconstruction where the researchers used a CNN-based feature extractor, none of the previous work has explored the potential

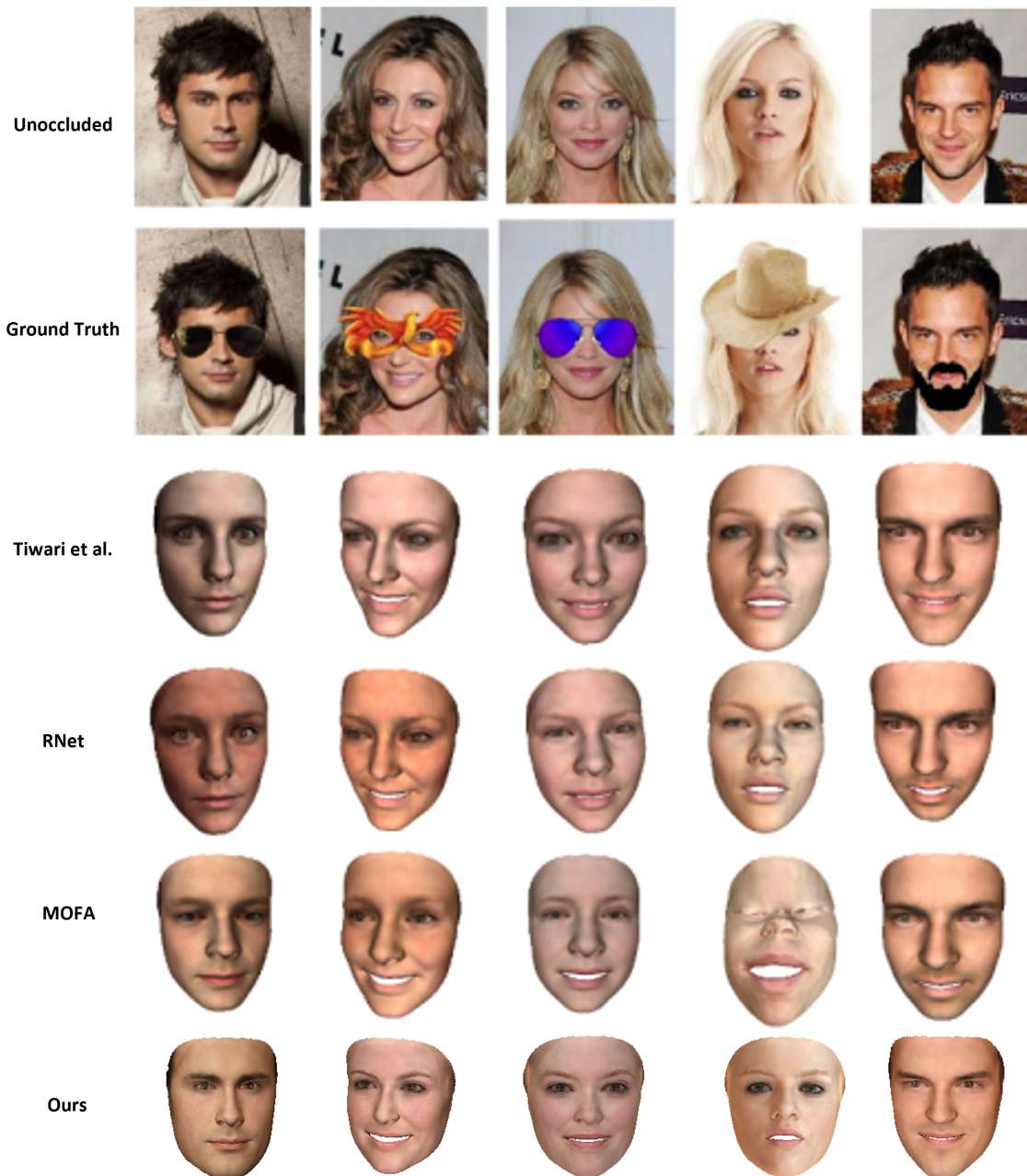


Fig. 4.2 Qualitative comparison of shape and texture on occluded images as ground truth with previous work - Tiwari et al. [133], Deng et al. [40], MOFA [130]. The results of the previous works are taken from Tiwari et al. [133]

of transformer networks. To mitigate one of the major drawbacks of transformer networks, i.e., their failure to extract local features, we introduce hierarchical feature extraction and aggregation of those multi-scale features. To explore the effectiveness of the transformers, we have conducted a detailed ablation study varying the backbones with different versions of

Table 4.2 Subjective evaluation results in four different hypotheses - Realism, texture, shape reconstruction, occlusion resistance. The table shows the Mean Opinion Score (MOS) and the standard deviation (Std.)

Methods	Realism		Texture-Completion		Shape-Reconstruction		Occlusion-Resistant	
	MOS	Std.	MOS	Std.	MOS	Std.	MOS	Std.
GANFIT [57]	3.53	0.49	3.73	0.57	3.26	0.44	-	-
Genova et al. [58]	3.13	0.61	3.06	0.24	-	-	-	-
Tran et al. [134]	2.33	0.69	2.4	0.48	2.93	0.44	-	-
Deng et al [40].	3.26	0.44	3.46	0.61	3.13	0.49	3.0	0.36
Tewari et al. [129]	2.73	0.57	2.66	0.47	2.6	0.48	2.13	0.49
Tiwari et al. [133]	-	-	-	-	-	-	3.2	0.4
Ours	3.6	0.48	3.66	0.47	3.2	0.54	3.4	0.48

pretrained (with 1K and 22K of ImageNet data variants) Swin Transformers - tiny, small, base, and large. We also compared their performance with other CNN-based backbones like Resnet, EfficientNet, and other Vanilla Transformers. We have published the benchmark results with their computational complexity, like a number of parameters and GFLOPs. We have made interesting observations where EfficientNet performs better than the vanilla ViTs, and the Swin-Base backbone is able to outperform all of these backbones with a comparatively small amount of parameters and computational complexity. We have also performed an ablation study to see the individual influence of the different loss functions on the overall loss. As the landmark loss mostly helps the network to scale the face properly and learn the alignment, we have put comparatively smaller weight on that. Through our extensive study, we have found that unsupervised photometric loss and supervised shape loss play the most important role in overall network training.

We evaluated our model on two aspects - 3D face reconstruction and 3D face alignment - across two popular evaluation datasets, AFLW2000-3D [168], and MICC Florence [8]. In the face alignment task, compared with ten previous works, our work achieves comparable results with SADRNet [116] and 3DDFAv2 [64] while outperforming the other works in quantitative evaluation. While comparing qualitatively, we have found our model produces good results compared to others. Particularly, for samples that have partial occlusion, our method performs well, while others output larger errors. When evaluated for the face reconstruction task, we followed the evaluation protocol provided by GANFIT [57]. From the comparative results, we have found that our work outperforms all the previous works by a reasonable margin except the GANFIT [57] method by producing a smaller mean error on the face shape. Figure 4.1 shows a qualitative comparison of the generated face shape on MoFA

[130] test dataset. We also test our method against the faces with occlusions. We believe that as we have put a higher weight on the face-recognizer-based perceptual error, our method provides high-quality results in terms of textures in the high occlusion cases and is able to preserve the identity information better than the previous works. Figure 4.2 provides some comparative results on the occluded faces which clearly shows that our method preserves the identity features (like face textures and colors) better than the other methods. To justify the qualitative evaluation, we conducted a subjective study with fifteen participants. We compiled the response and computed the mean opinion score (MOS) in terms of realism, texture completion, shape reconstruction, and occlusion resistance. Table 4.2 shows the comparative MOS scores on those above factors. From the output scores, we have found that our method gives the best results for realism and occlusion resistance while coming second best in texture and shape reconstruction.

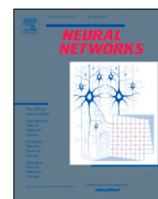
In summary, through this work, we have introduced the vision transformer in the face reconstruction task from a single-face image. We have proposed a hierarchical feature fusion mechanism to learn the local features as well as the long-distance dependencies through the transformers. We have published a new benchmark based on different backbones of ViTs in single image-based face reconstruction. Though through extensive ablation studies, we have found that transformer-based methods are able to achieve near SOTA performance compared to the traditional convolution methods, a major drawback of these methods is their huge model size and high computational cost. Even the convolution-based SOTA methods are also computationally expensive, which makes these methods unsuitable for edge devices. Removing the statistical model dependency and estimating dense 3D face landmarks can be a solution to reduce the network size and computational cost while helping to predict the face geometry. We will discuss this approach in the next chapter.

4.5 Copy of Published Works



Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

3D face-model reconstruction from a single image: A feature aggregation approach using hierarchical transformer with weak supervision[☆]

Shubhajt Basak^{a,*}, Peter Corcoran^b, Rachel McDonnell^c, Michael Schukat^a^a School of Computer Science, National University of Ireland Galway, Galway H91 TK33, Ireland^b Department of Electronic Engineering, College of Science and Engineering, National University of Ireland Galway, Galway H91 TK33, Ireland^c School of Computer Science and Statistics, Trinity College Dublin, Dublin 2, Ireland

ARTICLE INFO

Article history:

Received 4 June 2022

Received in revised form 24 August 2022

Accepted 19 September 2022

Available online 1 October 2022

Keywords:

Face reconstruction

Hierarchical transformer

Feature fusion

ViT

Swin Transformer

ABSTRACT

Convolutional Neural Networks (CNN) have gained popularity as the de-facto model for any computer vision task. However, CNN have drawbacks, i.e. they fail to extract long-range perceptions in images. Due to their ability to capture long-range dependencies, transformer networks are adopted in computer vision applications, where they show state-of-the-art (SOTA) results in popular tasks like image classification, instance segmentation, and object detection. Although they gained ample attention, transformers have not been applied to 3D face reconstruction tasks. In this work, we propose a novel hierarchical transformer model, added to a feature pyramid aggregation structure, to extract the 3D face parameters from a single 2D image. More specifically, we use pre-trained Swin Transformer backbone networks in a hierarchical manner and add the feature fusion module to aggregate the features in multiple stages. We use a semi-supervised training approach and train our model in a supervised way with the 3DMM parameters from a publicly available dataset and unsupervised training with a differential renderer on other parameters like facial keypoints and facial features. We also train our network on a hybrid unsupervised loss and compare the results with other SOTA approaches. When evaluated across two public datasets on face reconstruction and dense 3D face alignment tasks, our method can achieve comparable results to the current SOTA performance and in some instances do better than the SOTA methods. A detailed subjective evaluation also shows that our method performs better than the previous works in realism and occlusion resistance.

© 2022 Elsevier Ltd. All rights reserved.

1. Introduction

Retrieving the 3D face shape geometry from a single 2D facial image is an important problem in computer vision research. It has a wide range of applications in face analysis (Garrido et al., 2015; Thies, Zollhofer, Stamminger, Theobalt, & Nießner, 2016), facial expression estimation (Bejaoui, Ghazouani, & Barhoumi, 2017), face manipulation (Geng, Cao, & Tulyakov, 2019; Shu et al., 2017), 3D face recognition (Echeagaray-Patron, Kober, Karnaukhov, & Kuznetsov, 2017; Tu et al., 2020; Zhao et al., 2018), facial animation (Cudeiro, Bolkart, Laidlaw, Ranjan, & Black, 2019; Karras, Aila, Laine, Herva, & Lehtinen, 2017) etc. With the advancement of deep learning methods, estimating the accurate 3D face shape from a 2D image without any 3D labels in an unsupervised or

semi-supervised way has become very popular in the current research (Deng, Yang, et al., 2019; Genova et al., 2018; Tewari et al., 2018; Tran & Liu, 2018; Wu, Rupprecht, & Vedaldi, 2020). Most of these methods apply an analysis-by-synthesis method to learn a non-linear 3D face model trained on a large set of unlabeled RGB face image data by fitting a 3D Morphable Model (3DMM) first introduced by Blanz and Vetter (1999). These methods use a convolutional neural network (CNN)-based encoder network to learn the scene illumination, projection, shape, and albedo parameters, and a decoder network to map the learned non-linear 3DMM parameters to the 3D face. A differentiable rendering layer is added to the pipeline to train the decoder by minimizing the difference between the ground truth face image and the reconstructed face.

Though these CNN-based 3DMM feature extractor methods show very good results, by its fundamental design, convolutions are local operations and CNNs are sometimes incapable to process global information. Skip-connections can overcome this shortcoming, but sometimes fail to extract fine spatial information because of the misalignment of different layer features. Recently,

[☆] This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224.

* Corresponding author.

E-mail address: s.basak1@nuigalway.ie (S. Basak).

transformers have gained popularity and replaced the traditional CNN-based approach to overcome its shortcomings, due to their ability to capture long-term dependencies. Specifically, vision transformers (ViTs) (Dosovitskiy et al., 2020) have achieved state-of-the-art (SOTA) results in many computer vision tasks like image classification (Chen, Fan, & Panda, 2021), object detection (Sheng et al., 2021), and image segmentation (Zheng et al., 2021) due to their capability to extract global context features.

In this work, we leverage this ability of the transformer and apply it to the face reconstruction task. As per our assessment, none of the previous 3D face reconstruction work examined the power of the vision transformer. To avoid the large memory consumption of traditional transformers we use the Swin Transformer (Liu et al., 2021) as the backbone of the feature extractor, which reduces the computational cost and makes the model size smaller compared to other vanilla transformers. To add emphasis on the local feature extraction, we also integrate it with a hierarchical feature aggregation module, which fuses the multi-scale features and performs coarse-to-fine feature extraction. This further improves the network performance. We take a semi-supervised approach to train our network on unlabeled RGB face images through the differential rendering method and labeled ground truth face images with their corresponding 3DMM parameters, optimizing a hybrid loss function. We evaluate our model on two evaluation tasks, 3D face reconstruction and 3D dense face alignment with two public datasets. The quantitative experimental results show that our method achieves comparable results to the SOTA and in a few instances outperforms the current SOTA. We have also conducted a detailed subjective evaluation to compare our work with the previous works in terms of overall realism, shape and texture reconstruction, and performance of the model against occlusions. The results show that our work performs better in realism and occlusion resistance. We have done a study to compare our work with other feature extractor backbones including the convolution networks and the vanilla vision transformers and presented the results. Additionally, we perform an extensive set of ablation studies to investigate the performance while varying the feature fusion, Swin Transformer complexities, and multi-loss functions, and present the results in Section 6.

The rest of this paper is organized as follows: Section 2 presents a review of the related literature on monocular face reconstructions and visual transformers. The building blocks of our model, including the 3DMM face model, camera model, scene lighting, rendering, and the hierarchical Swin Transformer with the Feature Aggregation Module (FAM) are explained in Section 3. The training methodology and multi-loss functions are described in Section 4. Section 5 provides details of the experiments including the training datasets. It also provides the evaluation results and the test dataset descriptions. Section 6 presents a detailed ablation study on the effects of different Swin Transformer models, feature extractor backbones, feature aggregations, and multi-loss functions. Finally, the limitations and scope of improvements are discussed in Section 7 followed by the conclusion in Section 8.

2. Related works

3D face reconstruction from a monocular face image is a complex task because of the lack of 3D information present in a 2D image. It requires prior knowledge to resolve it. Statistical 3D face models are one of the most mentioned ways in literature to add this prior knowledge. Due to advancements in deep learning, some model-free methods are also proposed that regress the 3D shape from a single image without any prior statistical parametric models. In this section, we will discuss the current advancement in the 3D face reconstruction task and the use of feature aggregation with Swin Transformers in computer vision tasks.

2.1. Model based face reconstruction

Statistical Face Models: The most common and popular 3D face model used is the 3D Morphable Model (3DMM) proposed by Blanz and Vetter (1999). It consists of a shape and albedo (texture or color) model learned from a Principle Component Analysis (PCA). Basel Face Model (BFM) (Paysan, Knothe, Amberg, Romdhani, & Vetter, 2009) is another popular 3DMM face model constructed by applying a non-rigid iterative closest point (NICP) algorithm that decomposes the expression bases from the shape bases. SFM (Surrey Face Model) (Huber et al., 2016) is built using dense correspondence via an iterative multi-resolution dense 3D registration method, which has a diverse variation in age and ethnicity. In later years other works evolved that built 3D face modes on top of these early works. Face datasets like the Large Scale Facial Model (LSFM) (Booth, Roussos, Zafeiriou, Ponniah, & Dunaway, 2016), Facewarehouse (Cao, Weng, Zhou, Tong, & Zhou, 2013) and FLAME (Li, Bolkart, Black, Li, & Romero, 2017) are some of the multi-linear or bi-linear face models with additional attributes for identity and expressions. Ranjan, Bolkart, Sanyal, and Black (2018) created the non-linear face model COMA through a deep learning-based autoencoder model. A more detailed analysis of the 3DMM evolution can be found in Egger et al. (2020).

Methods based on Optimization: These methods based on optimization iteratively try to fit the 3DMM models to an image, video, or collection of images (Blanz, Basso, Poggio, & Vetter, 2003; Blanz & Vetter, 1999; Fried, Shechtman, Goldman, & Finkelstein, 2016; Roth, Tong, & Liu, 2016). More specifically, as these try to align the generated images from the 3DMM with the image based on image features such as landmarks, their performance drops with occlusions in faces.

Methods based on Deep Learning: With the advancement of deep learning, face reconstruction using deep neural networks became popular. These methods, which try to regress the 3DMM parameters, mainly depend on face autoencoders (Deng, Yang, et al., 2019; Gao et al., 2020; Genova et al., 2018; Richardson, Sela, & Kimmel, 2016; Tuan Tran, Hassner, Masi, & Medioni, 2017) which learn the latent distribution of the face with the statistical parameters, and connect them with the renderer to complete the end-to-end training. Guo, Cai, Jiang, Zheng, et al. (2018), Tewari et al. (2018), Tráň et al. (2018) proposed a coarse-to-fine strategy with a coarse linear network to learn the 3DMM parameters and a fine scale network for further corrections. These methods are highly constrained by the initial base face shape generated from the linear 3DMM.

To overcome this limitation later work proposed nonlinear models. Gao et al. (2020) used an encoder network to learn the pose, identity, expression, and lighting features and introduced the non-linearity in the decoder by using a discriminator, which forces the decoder to learn face shapes that follow the distribution of real faces. Deng, Yang, et al. (2019) used multiple view similarity and recovered the final face reconstruction by combining the single-view reconstructions according to confidence scores. Zhou, Deng, Kotsia, and Zafeiriou (2019) used colored mesh decoding to represent the non-linear 3DMM models. Ranjan et al. (2018) learned the 3D face shape using spectral graph convolution networks. Some of the more recent work (Guo, Yu, Lattas, & Deng, 2022) propose a simultaneous reconstruction of the face in world space and predict face landmarks in 2D image plane to improve the results under perspective projection (e.g. — the face is very close to the camera). Zielonka, Bolkart, and Thies (2022) propose MICA (Metric fAce) and introduced a metrical benchmark to measure the absolute error with respect to the face reconstruction task.

2.2. Swin transformer with feature fusion

The transformer architecture was originally proposed for natural language processing (NLP) tasks as it can extract global context features and model long-range dependency effectively. It is made of multiple stacked encoder/decoder layers with a self-attention mechanism embedded in it. To further improve its results, a multi-head attention mechanism (Vaswani et al., 2017) was proposed to calculate the attention among different positions jointly. In recent years transformers have shown better results than traditional CNNs in different vision tasks such as image classification, image segmentation, and object detection as discussed in Section 1. It splits the images into multiple patches and applies linear embeddings to the individual patches before sending them through the transformers as tokens (Dosovitskiy et al., 2020). The Swin Transformer (Liu et al., 2021) reduces the complexity of the traditional transformers from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ by limiting the self-attentions within the non-overlapping window, instead of calculating the global attention between all tokens. Following the pioneering work of Lin et al. (2017), which proposed the Feature Pyramid Networks for object detection, similar approaches have been used in some of the computer vision tasks which fuse the hierarchical feature fusion with the transformer architecture. For example, Zhu et al. (2022) proposed a depth supervised salient object detection using the Swin Transformer backbone and hierarchical feature aggregation.

3. Background and preliminaries

This section describes the basics of the different modules and models used in this work. We use the 3DMM-based Basal Face Model (BFM) as the parametric face model. As a prerequisite of our rendering pipeline, we regress the face pose and scene illumination. The 3DMM, face pose, and scene illumination parameters are generated by the Swin Transformer framework. We detail these sub-modules as follows:

3.1. 3DMM face model

In a 3DMM face model the 3D face shape $S \in \mathbb{R}^{3N \times 1}$ with N vertices and the face texture $T \in \mathbb{R}^{3N \times 1}$ is defined through the following equations:

$$S = S(\alpha_{id}, \alpha_{exp}) = \bar{S} + B_{id}\alpha_{id} + B_{exp}\alpha_{exp} \quad (1)$$

$$T = T(\alpha_{tex}) = \bar{T} + B_{tex}\alpha_{tex} \quad (2)$$

where $\bar{S} \in \mathbb{R}^{3N \times 1}$ and $\bar{T} \in \mathbb{R}^{3N \times 1}$ are the mean shape and texture respectively. $B_{id} \in \mathbb{R}^{3N \times K}$ are the first K principle components trained on facial scans with neutral expressions, $B_{exp} \in \mathbb{R}^{3N \times L}$ are the first L principle components trained on a predefined offset of neutral scan and expression scans, $B_{tex} \in \mathbb{R}^{3N \times M}$ are the first M principle components trained on facial texture. $\alpha_{id} \in \mathbb{R}^{K \times 1}$, $\alpha_{exp} \in \mathbb{R}^{L \times 1}$, $\alpha_{tex} \in \mathbb{R}^{M \times 1}$ are their corresponding coefficient vectors which are being learned by the regressor network developed with the Swin transformer to generate the 3D face. Similar to Deng, Yang, et al. (2019), we use the Basel Face Model (Paysan et al., 2009) as the base 3DMM model, which has the non-trainable parameters \bar{S} , \bar{T} , B_{id} , B_t set in it. For B_{exp} we use similar to Deng, Yang, et al. (2019) the data from the work of Guo et al. (2018), which has been trained from Facewarehouse (Cao et al., 2013). We learn the 3DMM feature vectors excluding the neck and ear region following Deng, Yang, et al. (2019). The final dimensions of the three parameters are $\alpha_{id} \in \mathbb{R}^{80}$, $\alpha_{exp} \in \mathbb{R}^{64}$ and $\alpha_{tex} \in \mathbb{R}^{80}$. The resulting 3DMM model consists of 35 709 vertices and 70 789 faces.

3.2. Camera model

Similar to previous work, (Deng, Yang, et al., 2019; Gao et al., 2020) we use the perspective camera model. The focal length is selected empirically as in Deng, Yang, et al. (2019). The face pose is obtained through the 3D–2D projection geometry with its rotation (yaw, pitch and roll) \mathbb{R}^3 and translation (x, y, z shift) \mathbb{R}^3

3.3. Scene illumination model

We approximate the scene illumination using Spherical Harmonics (SH) (Ramamoorthi & Hanrahan, 2001), while the 3D faces are assumed to be a Lambertian surface. Similar to Deng, Yang, et al. (2019) we have chosen the illumination model where the radiosity of a vertex v_i with surface normal n_i and texture t_i can be calculated as

$$C(n_i, t_i | \delta) = t_i \cdot \sum_{b=1}^{B^2} \delta_b \phi_b(n_i) \quad (3)$$

where ϕ_b is the SH coefficients of the SH basis function $\delta_b : \mathbb{R}^3 \rightarrow \mathbb{R}$. We have also set the illumination as white light with 3 bands (Tewari et al., 2017) such that $\delta \in \mathbb{R}^9$.

3.4. Rendering layer

As we do not have any ground truth 3D face scans, we utilize a differential rendering layer to render the predicted 3DMM models. The 3D model is projected into an image plane with a weak perspective projection which follows:

$$S_{2D} = \rho * P_r * R * S + t_{2D} \quad (4)$$

where $S_{2D} \in \mathbb{R}^{2 \times N}$ is the face shape projected on the image plane. $P_r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ is the orthographic projection matrix, and R is the rotation matrix in Euler angles for yaw, pitch and roll. $t_{2D} = [t_x, t_y]^T$ is the image plane translation vector and ρ is the scale factor. As stated in the previous section, we use the Lambertian surface and spherical harmonics illumination with three bands with the illumination parameter δ . The rendering process is a function of $\chi = \{\alpha_{id}, \alpha_{exp}, \alpha_{tex}, yaw, pitch, roll, \delta, \rho, t_{2D}\}$. The rendering layer is implemented with the help of an open-source differential rendering library called Nvdiffrast (Laine et al., 2020).

3.5. Swin transformer layer

To learn the 3DMM parameters during the feature extraction, we use multiple Swin Transformer Layers (STL), which replace the traditional convolution layers. The STL is constructed based on the original transformer layer used in Natural Language Processing tasks. Instead of the global self-attention used by this conventional transformer, Swin uses the self-attention within the non-overlapping local windows for fast computation and efficient modeling. To achieve cross-window connections and long-range dependencies, a shifted window partitioning mechanism is added as well. Thus it achieves better performance for different pixel-wise computer vision tasks.

The input to the Swin block is a token $X \in \mathbb{R}^{H \times W \times D \times C}$ with a patch resolution of H' , W' , D' and the dimension of $H' \times W' \times D' \times C'$, where H , W , D and S are the image height, width, depth, and sample size respectively. As stated in the original work, we use a patch partition layer to make a sequence of 3D tokens that have dimensions of $\left[\frac{H}{H'}\right] \times \left[\frac{W}{W'}\right] \times \left[\frac{D}{D'}\right]$ and project them into a C' -dimensional embedding layer. For efficient token interaction, self-attention is computed in non-overlapping windows, which are created during the partitioning stage. A $(M \times M \times M)$ window

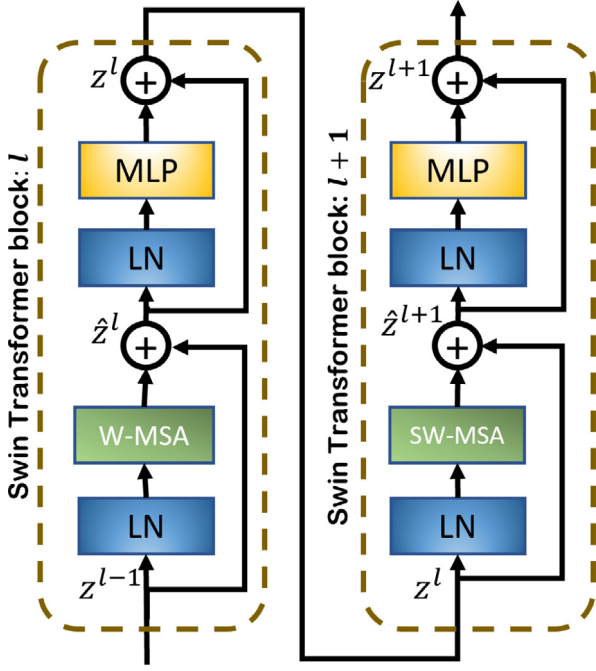


Fig. 1. Representation of a Swin Transformer block.

is used to evenly partition the 3D tokens into $\left[\frac{H'}{M}\right] \times \left[\frac{W'}{M}\right] \times \left[\frac{D'}{M}\right]$ regions in a given layer l . In layer $l + 1$ the window is shifted by $\left[\frac{M}{2}\right], \left[\frac{M}{2}\right], \left[\frac{M}{2}\right]$ voxels. The outputs of the layer l and $l + 1$ in the STL can be calculated as follows:

$$\begin{aligned} \hat{z}^l &= \text{W-MSA}(\text{LN}(z^{l-1})) + z^{l-1} \\ z^l &= \text{MLP}(\text{LN}(\hat{z}^l)) + \hat{z}^l \\ \hat{z}^{l+1} &= \text{SW-MSA}(\text{LN}(z^l)) + z^l \\ z^{l+1} &= \text{MLP}(\text{LN}(\hat{z}^{l+1})) + \hat{z}^{l+1} \end{aligned} \quad (5)$$

Fig. 1 shows two consecutive Swin Transformer blocks represented by Eq. (5). Here the LN stands for the linear layer, and MLP denotes the multi-layer perceptron with the Gaussian Error Linear Unit (GELU) activation function. The standard multi-head self-attention layer used in a normal transformer is replaced by the window-based multi-head self-attention (W-MSA) and the shifted window-based multi-head self-attention (SW-MSA) respectively. For the efficient computation of the shifted windows task, we adopted the 3D cyclic-shifting as stated in the original work (Liu et al., 2021). The self-attention is been computed as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (6)$$

where Q, K, and V are queries, keys, and values respectively, while d is the dimension of the query and key.

3.6. Hierarchical feature extraction

The regressor part of the network as shown in Fig. 2 extracts hierarchical features in four different scales at four stages. It starts from an input image and gradually merges neighboring image patches while progressing to deeper layers. The input RGB image is first divided into sizes of 4×4 non-overlapping patches with the final patch dimension of $4 \times 4 \times 3 = 48$. Then a linear embedding layer projects this feature into an arbitrary dimension denoted as C , resulting in a patch token with the shape of $\left(\frac{H}{4} \times \frac{W}{4}, C\right)$. To generate the hierarchical features in the later stage, the patches are merged by concatenating each 2×2

group of neighboring patches and passing the result through a linear layer to get a high-dimensional patch. If N is the number of tokens and D denotes the input dimension, the output shape of the patch merging is $\left(\frac{N}{4}, 2D\right)$. At each stage, a sequence of STB is applied to merge the patches while keeping the number of tokens unchanged. The four stages have 2, 2, 18, and 2 STB respectively. Each stage produces 4 hierarchical features denoted as F_4, F_3, F_2, F_1 with shapes $\left(\frac{H}{4} \times \frac{W}{4}, C\right), \left(\frac{H}{8} \times \frac{W}{8}, 2C\right), \left(\frac{H}{16} \times \frac{W}{16}, 4C\right), \left(\frac{H}{32} \times \frac{W}{32}, 8C\right)$ respectively.

3.7. FAM: Multi-scale feature aggregation module

As discussed in the previous Section 3.6, the regressor gets the hierarchical features in four different spatial resolutions. However, using only the hierarchical regressor results in a large gap in semantics because of the four different learning stages. The high-resolution feature maps have very detailed low-level features, but fail to capture salient characteristics. On the other hand, the low-resolution maps capture semantically significant high-level features only. In order to utilize both high and low-level features for a dense prediction similar to DFTR (Zhu et al., 2022), we propose a multi-scale feature aggregation module (FAM) that will gradually aggregate the features in different stages.

Fig. 2b shows the detailed architecture of the module. It takes the low and high-resolution feature maps as input and aggregates them to get an output with the same shape as the high-resolution map. The coarse feature (f_{1in}) is passed through a bi-linear up-sampling layer to match the spatial dimension of the input with the high-resolution map ($4N, 2D$) through interpolation. The high-resolution feature map (f_{2in}) is passed through a convolution and a linear layer to enlarge its channel dimension by a factor of two ($4N, 2D$). Both the outputs are then passed through a multiplication and a channel-wise concatenation layer. The multiplication operation enhances the common pixels and reduces the effect of the ambiguous pixels. The output f_{mid} with dimension $(4N, 6D)$ can be represented as :

$$\begin{aligned} f_{mid} &= U(f_{1in}) \oplus \\ &L_{\theta_2}(\text{Conv}_{\theta_1}(f_{2in})) \oplus \\ &(U(f_{1in}) \otimes L_{\theta_2}(\text{Conv}_{\theta_1}(f_{2in}))) \end{aligned} \quad (7)$$

Finally, the output f_{mid} is again passed through a convolution and a linear layer to get the desired output f_{out} with a reduced channel dimension of $(4N, D)$,

$$f_{out} = L_{\theta_4}(\text{Conv}_{\theta_3}(f_{mid})) \quad (8)$$

where \oplus and \otimes are the concatenation and the multiplication operation respectively. U, L and Conv are the up-sampling, Linear and Convolution layer respectively and $\theta_1, \theta_2, \theta_3, \theta_4$ are their trainable parameters.

4. Methodology

We have designed a regressor network that performs end-to-end adversarial training to extract the disentangled semantic features of a human face. Similar to Gao et al. (2020), we incorporate an inverse rendering method that uses a parameterized illumination model and a differentiable renderer to programmatically render back the 2D face image from the 3D face parameters, thereby varying the identity, illumination, expression, pose, and texture.

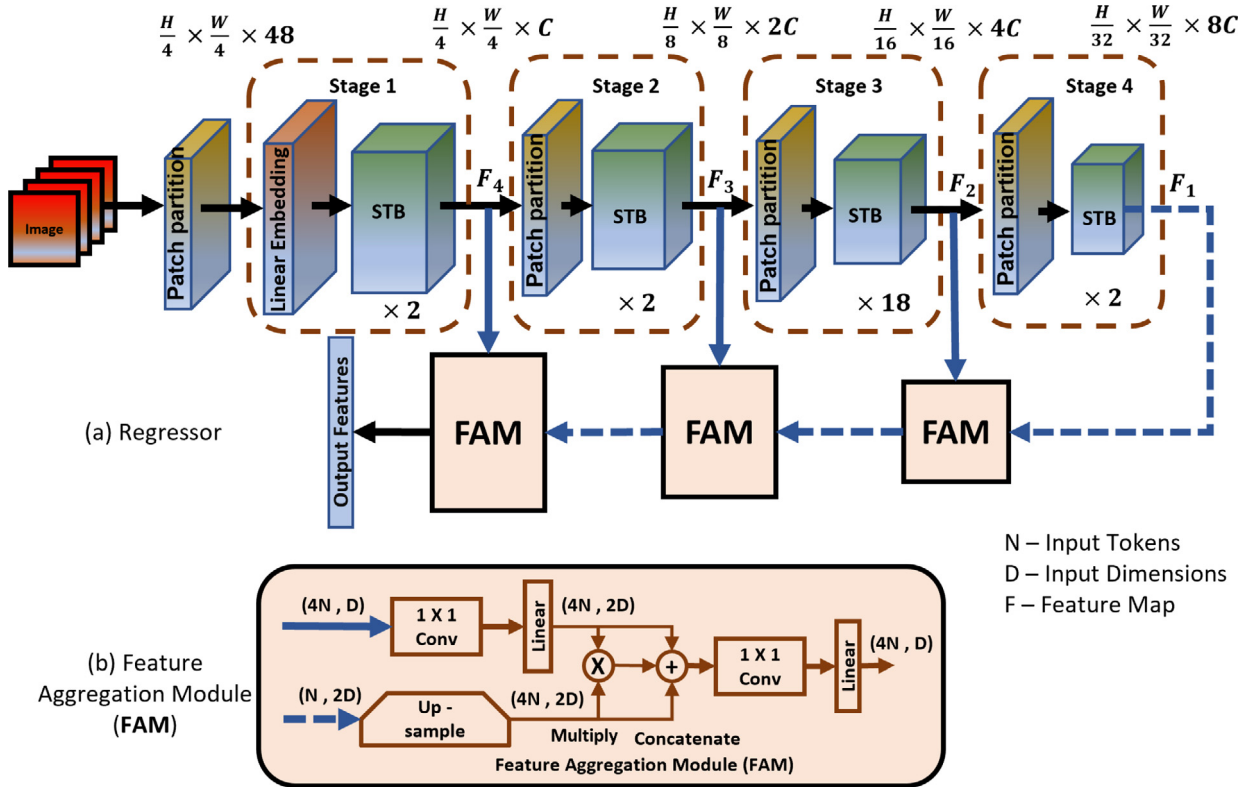


Fig. 2. (a) Overview of the network structure which consists of Swin Transformer Blocks (STB) in a hierarchical way. (b) FAM: Multi-scale feature aggregation module.

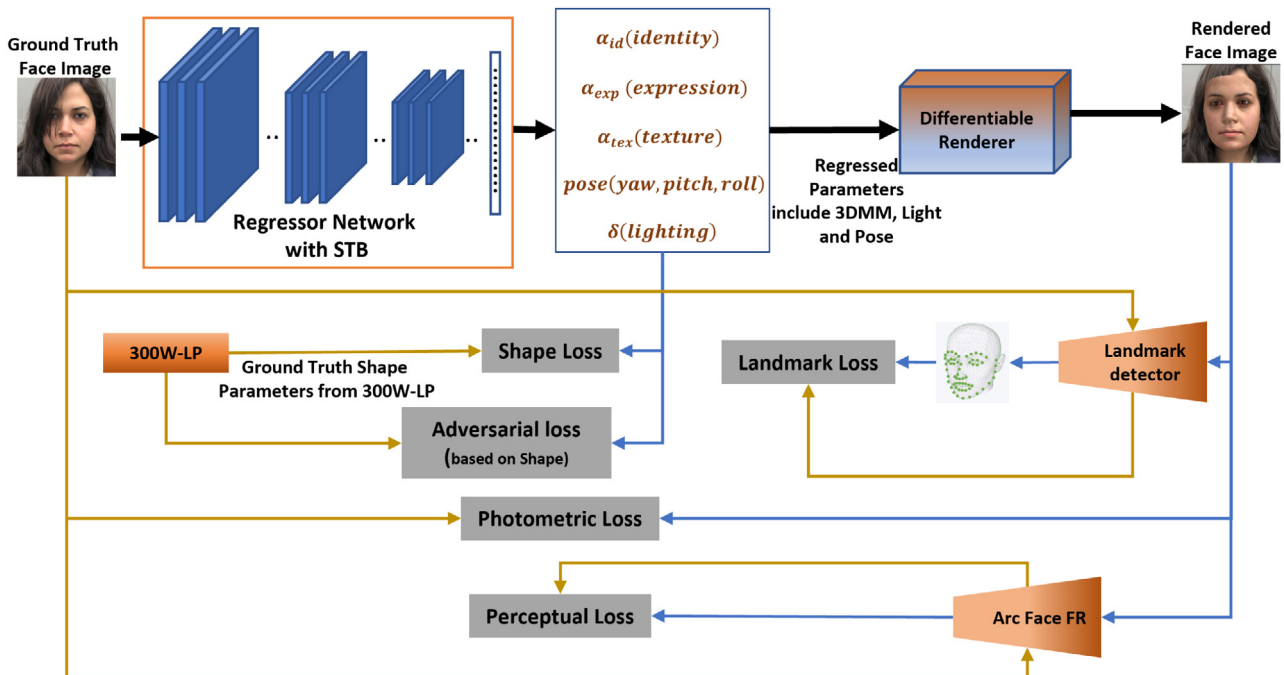


Fig. 3. Overview of the training pipeline with the five different loss functions and the differential renderer in place. The regressor network consists of the STB and FAM module.

4.1. Training pipeline

The overall training framework is comprised of a feature extractor that extracts the 3DMM parameters, and a differentiable

renderer that renders a synthetic face image out of that parameters, as shown in Fig. 3. We use a dual training methodology, where the input image set is composed of unlabeled face images for unsupervised training and a set of labeled face images that have the 3DMM parameters as described in Eq. (1) as the ground

truth. The regressor network comprises of the feature extractor with the FAM module as discussed in Sections 3.6 and 3.7. It regresses the 3DMM parameters α_{id} , α_{exp} , α_{tex} , pose (yaw, pitch, roll) and the illumination parameter δ . These parameters are then sent to a differential renderer to get the output face image and merged with the background using the face mask calculated by the regressor.

4.2. Loss functions

We have incorporated a hybrid loss similar to Gao et al. (2020) to train the network in a semi-supervised way. The overall loss function is defined as:

$$L = w_1 L_{lmk} + w_2 L_{pht} + w_3 L_{perc} + w_4 L_{shape} + w_5 L_{adv} \quad (9)$$

where L_{lmk} , L_{pht} , L_{perc} , L_{shape} , L_{adv} are the landmark loss, photometric loss, perceptual loss, shape loss and adversarial loss respectively with their corresponding weights w_1, w_2, w_3, w_4, w_5 . The different losses and setting the values of their corresponding weights are discussed in subsequent sections in detail.

4.2.1. Landmark loss

We calculate the landmark loss similar to Deng, Yang, et al. (2019), Gao et al. (2020) to capture the low-level information of the constructed face. The landmark positions on the 2D image domain provide weak supervision while training the network. We first run the state-of-the-art face alignment network (Bulat & Tzimiropoulos, 2017) with the ground truth images to get the 68 key points $\{q_n\}$ of the training faces. While training the network, we project the 3D landmark vertices of the reconstructed face shapes on the 2D face image to get the corresponding 2D landmarks $\{q'_n\}$. The loss function is calculated as follows:

$$L_{lmk}(x) = \frac{1}{N} \sum_{n=1}^N \omega_n \|q_n - q'_n(x)\|^2 + L_{gdl, lmk} \quad (10)$$

Here $\|\cdot\|$ is the l_2 norm, N is the number of keypoints, ω_n is the landmark weight, which is set to 20 for the inner mouth and nose and 1 for others (Deng, Yang, et al., 2019). We have also added the gradient difference loss (GDL) (Mathieu, Couprie, & LeCun, 2015) similar to Gao et al. (2020) denoted as $L_{gdl, lmk}$, which is applied on the sparse landmarks. It helps to maintain consistency of the distances between the different landmark points such as the upper and lower eyelids and upper and lower lips, therefore giving more weight to features like eye openings and mouth openings.

4.2.2. Photometric loss

A popular way to determine the difference between a ground truth image and the rendered image is to measure the photometric discrepancy. Since there are occlusions like hair which can degrade its performance, we first obtain a mask M using the work of Nirkin, Masi, Tuan, Hassner, and Medioni (2018) to get rid of the occlusions, before calculating the photometric loss as:

$$L_{pht} = M \odot (\|i' - i\|_2^2 + L_{gdl}) \quad (11)$$

Here also we have added the GDL (Mathieu et al., 2015) to reduce the pixel-wise discrepancies. \odot is the element-wise product function.

4.2.3. Perceptual loss

Training the network with the above-discussed photometric and landmark loss produces smooth textures and lower visual discrepancies, but the underlying 3D shapes are not learned properly. Therefore, similar to Deng, Yang, et al. (2019) we add a perception level loss to add the additional cues on shapes. The

intuition behind this loss is to extract the deep features from face images through a pre-trained face recognition (FR) model and try to minimize the cosine distance between the ground truth image features and the rendered image features. We have chosen ArcFace (Deng, Guo, Xue, & Zafeiriou, 2019) as the FR model which has the highest accuracy in FR tasks on popular public datasets like LFW 99.83% and YTF DB 99.02%. ArcFace has been trained on ResNet-100 [18] using the MS1M dataset (Guo, Zhang, Hu, He, & Gao, 2016) and further uses the additive angular margin loss to improve its result. The loss function is defined as:

$$L_{perc}(x) = 1 - \frac{\langle \text{arc}(i), \text{arc}(i'(x)) \rangle}{\|\text{arc}(i)\| \cdot \|\text{arc}(i'(x))\|} \quad (12)$$

where $\text{arc}(\cdot)$ is the features encoded by the ArcFace FR and $\langle \cdot, \cdot \rangle$ is the vector inner product.

4.2.4. Shape loss

The loss functions defined in the previous sections mostly help training the network in a semi-supervised way with unlabeled data or generated pseudo labels from pre-trained networks. To train the network with more shape cues we train the network in a supervised way with the help of labeled 3DMM parameters. We have used the 300W-LP dataset (Zhu, Lei, Liu, Shi, & Li, 2016) that has approximately 122k face images with its fitted 3DMM parameters across large poses created from a face profiling technique. Similar to Deng, Yang, et al. (2019) we have excluded the neck and ear of the BFM model, so our base 3D face template has 35 709 vertices. The L1 loss between the ground truth shape s is calculated from the 300W-LP database and the predicted shape s' parameters through Eq. (1). The shape loss is defined as:

$$L_{shape}(x) = \|s' - s[:, v]\|_1 \quad (13)$$

where v is the vertex indices of our base face template.

4.2.5. Adversarial loss

Though the above supervised training with the shape loss gives good results for the 300W-LP dataset, it fails to provide good results for some subsets of the unlabeled data which are used for the semi-supervised training. To keep the generated 3DMM parameter distribution near to the ground truth 3DMM parameters from 300W-LP, we have incorporated an adversarial training similar to Gao et al. (2020). Here a discriminator network is added at the end of the feature extractor that tries to discriminate the fake shapes reconstructed from the feature extractor network and the real shapes generated from 300W-LP dataset. We follow the Wasserstein Divergence GAN (Wu, Huang, Thoma, Acharya, & Van Gool, 2018) to get the min-max optimization as:

$$\min_G \max_D \mathbb{E}_{s' \sim \mathbb{P}_g} [D(s')] - \mathbb{E}_{s[:, v] \sim \mathbb{P}_r} [D(s[:, v])] - k \mathbb{E}_{\hat{s} \sim \mathbb{P}_u} [\|\nabla_{\hat{s}} D(\hat{s})\|^p] \quad (14)$$

where $-D(s')$ is the adversarial loss (L_{adv}). s' and s are the shape predicted by the network (fake shape) and the ground truth shape from 300W-LP (real shape) respectively with their probability distributions \mathbb{P}_g and \mathbb{P}_r . ∇ is the gradient operator and \mathbb{P}_u is the distribution derived from sampling uniformly the fake and real data along a straight line drawn between them.

4.2.6. Selection of weights

As we have used a combination of different losses as our objective function. We set the weights of each term of the loss function to balance the influence of each loss term. Following the previous works (Deng, Yang, et al., 2019; Gao et al., 2020), as the landmark loss only helps to align the generated face with the ground truth, to reduce its influence in other tasks, we set

a comparatively small weight for its loss term to 0.001 across all our experiments. We then conduct a large set of experiments as part of our ablation study to empirically set the rest of the weights. To observe the influence of the individual loss term we conducted experiments where we set the weights corresponding to photo-metric, perceptual, and shape loss individually to 1 and kept others to zero. Through the observation, we have found that photo-metric loss and shape loss has the highest influence on the result whereas putting higher values on the perceptual loss reduces the accuracy and results in degeneration of shape. So we have put a higher weight on the photo-metric and shape loss compared to the perceptual loss. In the final set of experiments, we have added a comparatively small weight to the adversarial loss to constrain the model towards the real 3DMM distributions during the supervised learning phase. The final values of the weights w_1, w_2, w_3, w_4, w_5 associated to the five loss functions landmark loss (L_{lmk}), photometric loss (L_{pht}), perceptual loss (L_{perc}), shape loss (L_{shape}), and adversarial loss (L_{adv}) are set to 0.001, 1.70, 0.30, 1.20, 0.10 respectively. The detailed results of the ablation experiments can be found in Table 6.

5. Experiments

In this section, we will discuss the experimental setup including the data preparation for both training and evaluation, and the training methodology of our proposed hybrid training, which includes both unsupervised training from unlabeled face images and supervised training on labeled 3DMM parameters. After which we will present the evaluation results both quantitative and qualitatively on dense face alignment and face reconstruction task. We will also conduct a detailed subjective evaluation of the face reconstruction task and share the results.

5.1. Training datasets

For the unsupervised and weakly supervised training we use in-the-wild face images from the two popular open face datasets: CelebA (Liu, Luo, Wang, & Tang, 2015) and VGGFace2 (Cao, Shen, Xie, Parkhi, & Zisserman, 2018). CelebA has a total of around 200k images with 10176 identities in it. VGGFace2 has 9131 subjects with around 3.31 million images. We use a subset of VGGFace2 which consists of approximately 300k images. The total collection of face images amounts to around 500k. We run Insightface (which is built on top of ArcFace (Deng, Guo, et al., 2019)) to clean the dataset and remove some of the ambiguous and extreme images, and the frames which contain multiple faces. For supervised training, we use the 300W-LP (Zhu et al., 2016) dataset, which has a total of 61 225 face images with 3837 identities. It provides the 68 3D face keypoints with the ground truth face images. During training, the input images are augmented through random scaling between [0.8, 1.0] and random horizontal flips on the go.

5.2. Implementation details

The complete network is trained in three steps through batch processing. At first, the datasets created from CelebA and VggFace2 are randomly split into two parts. Then the training process is as follows:

- In the first step the network is trained with the unlabeled face taken from the part 1 split of CelebA and VggFace2 with the three loss functions: Landmark Loss (L_{lmk}), Photometric Loss (L_{pht}) and Perceptual Loss (L_{perc}) for 100 epochs. We name this as unlabeled training.

- In the next step the model is further trained with the labeled 300W-LP dataset and the remaining second split of CelebA and VggFace2. The CelebA and VggFace2 data is passed through the Landmark Loss (L_{lmk}), Photometric Loss (L_{pht}) and Perceptual Loss (L_{perc}) as those do not have the ground truth shape parameters. The 300W-LP is trained on all five losses including the Shape Loss (L_{shape}) and Adversarial Loss (L_{adv}). We name this mixed training.

The effect of the individual loss and these two steps has been discussed in more detail in the ablation study section. For the feature extraction module, we have experimented with all the four available Swin Transformer models namely Swin Tiny (Swin-T), Swin Small (Swin-S), Swin Base (Swin-B), and Swin Large (Swin-L) with the different pre-trained weights based on the training on ImageNet-1k and ImageNet-22k datasets.

The input size of the face images is set to 224×224 , the first stage embedding dimension is chosen as $C = 192$ and the window size is 12. The number of heads and the number of blocks in the STB module for each stage of the regressor network is set to 6, 12, 24, 48 and 2, 2, 18, 2 respectively. The network is trained using the Adam Optimizer (Kingma & Ba, 2014) with an initial learning rate of $5e-5$ that is reduced by 10 every 50 epochs, and a batch size of 5. The experiments were carried out on an Intel Core i5-7400 3 GHz CPU with 32 GB RAM and an NVIDIA GeForce GTX TITAN X Graphical Processing Unit (GPU) with 12 GB of dedicated graphics memory.

5.3. Evaluation datasets

We evaluate our model on two aspects, 3D face reconstruction accuracy and dense face alignment. For face alignment, we chose the very popular evaluation dataset AFLW2000-3D (Zhu et al., 2016), and for 3D face reconstruction we use the MICC Florence (Bagdanov, Del Bimbo, & Masi, 2011) dataset.

- **AFLW2000-3D** is an in-the-wild face dataset with a large variation in illumination, pose, occlusion and expression. It has 2000 images with its 3DMM parameters to recover the ground truth face shape and the 68 3D face landmark points for face alignment. We use it for face alignment evaluation.
- **MICC Florence** is a 3D face dataset that has 3D mesh scanned by a structured-light system of 53 subjects and respective short video footage under three settings: 'cooperative', 'indoor' and 'outdoor'.

5.4. Evaluation on face alignment

We compare our work with the previous works in terms of the dense face alignment task both quantitatively and qualitatively.

5.4.1. Quantitative comparison

To measure the face alignment quantitatively we use the normalized mean error (NME) as the evaluation metric. NME is computed as the normalized mean Euclidean distance between each set of corresponding landmarks in the predicted result l and the ground truth l' :

$$\text{NME} = \frac{1}{N} \sum_{i=1}^N \frac{\|l_i - l'_i\|_2}{d} \quad (15)$$

Following the previous works (Ruan et al., 2021), the normalization factor d is computed as $\sqrt{h * w}$, where h and w are the height and width of the bounding box respectively. Similar to Feng et al. (2018), Ruan et al. (2021) for 2D and 3D sparse alignment we consider all 68 landmark points. We divide the dataset

Table 1

Comparative Results on AFLW2000-3D on the task of Sparse Alignment with 68 landmarks. The NME (%) is reported for different yaw angles and for a balanced subset with an average distribution of yaw angles. Results of the previous works are taken from the SADRNet (Ruan, Zou, Wu, Wu, & Wang, 2021) paper.

Method	(0°,30°)	(30°, 60°)	(60°, 90°)	Balanced
3DDFA	3.78	4.54	7.93	5.42
3DFAN	2.77	3.48	4.61	3.62
DeFA	–	–	–	4.50
3DSTN	3.15	4.33	5.98	4.49
NonLinear 3DMM	–	–	–	4.12
PRNet	2.75	3.51	4.61	3.62
DAMDN	2.90	3.83	4.95	3.89
CMD	–	–	–	3.90
SPDT	3.56	4.06	4.11	3.88
3DDFAv2	2.63	3.42	4.48	3.51
SADRNet	2.66	3.30	4.42	3.46
Ours	2.68	3.37	4.51	3.54

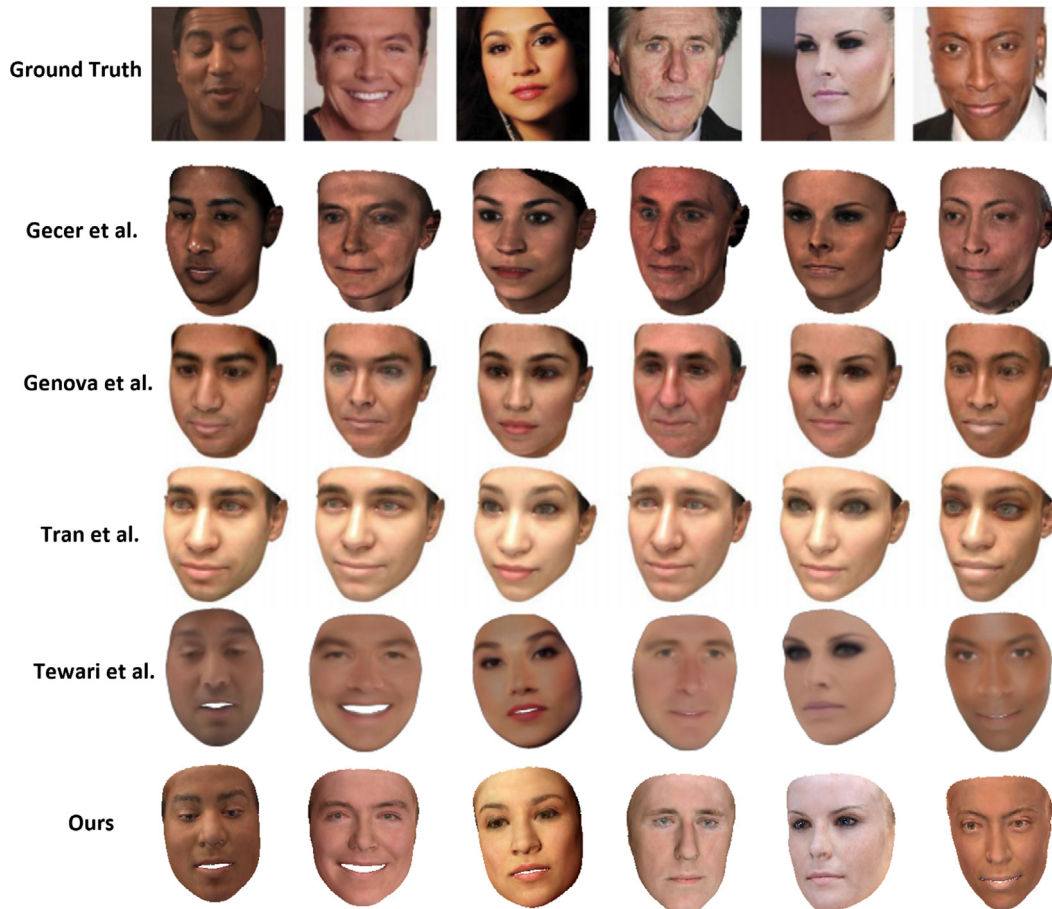


Fig. 4. Comparative results having rendered texture with previous works - Gecer, Ploumpis, Kotsia, and Zafeiriou, Genova et al., Tuan Tran et al., Tewari et al. Source: The images are taken from the paper of Gecer et al.

based on the yaw angles (0°, 30°), (30°, 60°) and (60°, 90°) and a balanced subset created by taking a random sample from the whole dataset. The results are presented in Table 1. We compare our outcome with previous works – 3DDFA (Zhu et al., 2016), 3DFAN (Bulat & Tzimiropoulos, 2017), DeFA (Liu, Jourabloo, Ren, & Liu, 2017), 3DSTN (Bhagavatula, Zhu, Luu, & Savvides, 2017), non-linear 3DMM (Tran & Liu, 2019), PRNet (Feng et al., 2018), DAMDN (Jiang, Wu, & Kittler, 2019), CMD (Zhou et al., 2019), SPDT (Piao, Qian, & Li, 2019), 3DDFA Ver 2 (Guo et al., 2020), and SADRNet (Ruan et al., 2021). It can be seen that our method achieves comparable results to SADRNet, 3DDFAv2 and beats the

other works in most of the measures. Particularly our method produces good results in balanced partitions.

5.4.2. Qualitative comparison

We also compare our results with previous works including MGCNet (Shang et al., 2020), PRNet (Feng et al., 2018) and SADRNet (Ruan et al., 2021). The results are presented in Fig. 5. MGCNet fits the shape and poses parameters on a 3DMM model learning from a CNN network. PRNet directly regresses the face mesh vertices and UV position maps. SADRNet decomposes the dense face alignment and face reconstruction task into several

Table 2

Quantitative comparison of the Florence MICC dataset on the task of face reconstruction. The table shows the mean error (Mean) and the standard deviation (Std.).

Method	Co-operative		Indoor		Outdoor	
	Mean	Std.	Mean	Std.	Mean	Std.
Tran et al.	1.93	0.27	2.02	0.25	1.86	0.23
Booth et al.	1.82	0.29	1.85	0.22	1.63	0.16
Genova et al.	1.50	0.13	1.50	0.11	1.48	0.11
Deng et al.	0.978	0.22	1.083	0.26	1.075	0.25
Gecer et al.	0.95	0.107	0.94	0.106	0.94	0.106
Ours	0.956	0.23	1.086	0.24	0.964	0.22

Table 3

Subjective evaluation results in four different hypotheses – Realism, texture, shape reconstruction, occlusion resistance. The table shows the Mean Opinion Score (MOS) and the standard deviation (Std.).

Methods	Realism		Texture-Reconstruction		Shape-Reconstruction		Occlusion-Resistant	
	MOS	Std.	MOS	Std.	MOS	Std.	MOS	Std.
Gecer et al.	3.53	0.49	3.73	0.57	3.26	0.44	–	–
Genova et al.	3.13	0.61	3.06	0.24	–	–	–	–
Tran et al.	2.33	0.69	2.4	0.48	2.93	0.44	–	–
Deng et al.	3.26	0.44	3.46	0.61	3.13	0.49	3.0	0.36
Tewari et al.	2.73	0.57	2.66	0.47	2.6	0.48	2.13	0.49
Tiwari et al.	–	–	–	–	–	–	3.2	0.4
Ours	3.6	0.48	3.66	0.47	3.2	0.54	3.4	0.48

- Each face mesh from both the predicted and the ground truth is cropped at a radius of 95 mm around the nose tip similar to previous works. [Deng, Yang, et al. \(2019\)](#), [Gecer et al. \(2019\)](#), [Genova et al. \(2018\)](#), [Tuan Tran et al. \(2017\)](#) to evaluate the face shape reconstruction of the inner facial mesh.
- Then for each frame the 3D mesh is predicted by the network and coarsely aligned with the corresponding ground truth scans with the help of the 68 landmark points.
- To get rid of any misalignment, a rigid Iterative Closest Point (ICP) algorithm ([Besl & McKay, 1992](#)) is applied without deforming the predicted meshes.
- Finally, the error is calculated as the mean symmetrical point-to-plane distance.

The quantitative results on face shape are given in [Table 2](#) with their mean errors (Mean) and corresponding standard deviations (Std.). Our result is compared with the previous works of [Booth et al. \(2017\)](#), [Deng, Yang, et al. \(2019\)](#), [Gecer et al. \(2019\)](#), [Genova et al. \(2018\)](#), [Tuan Tran et al. \(2017\)](#). The results of [Booth et al. \(2017\)](#), [Gecer et al. \(2019\)](#), [Genova et al. \(2018\)](#), [Tuan Tran et al. \(2017\)](#) are taken from the GANFIT ([Gecer et al., 2019](#)) paper and the result of [Deng, Yang, et al. \(2019\)](#) is calculated by running the published pre-trained model from their work following the above mentioned methodology. From the result, it can be seen that our method is able to outperform all the previous work except the GANFIT with a reasonable margin.

5.5.2. Qualitative comparison

We further compare our results qualitatively with previous works.

Comparison with MoFA test dataset: [Fig. 4](#) shows comparative results on textures and shape on some samples from the MoFA ([Tewari et al., 2017](#)) test dataset compared with works from Ganfit ([Gecer et al., 2019](#)), [Genova et al. \(2018\)](#), [Tuan Tran et al. \(2017\)](#), [Tewari et al. \(2018\)](#). The corresponding shapes are compared and presented in [Fig. 8](#). Here we add the shapes available from [Tran and Liu \(2018\)](#). In both cases, our results outperform most of the previous works. Also, the texture and shape reconstructions preserve the identity characteristics better than in the previous works.

Comparison with MICC dataset: We compare our network output on the publicly available MICC dataset with the previous

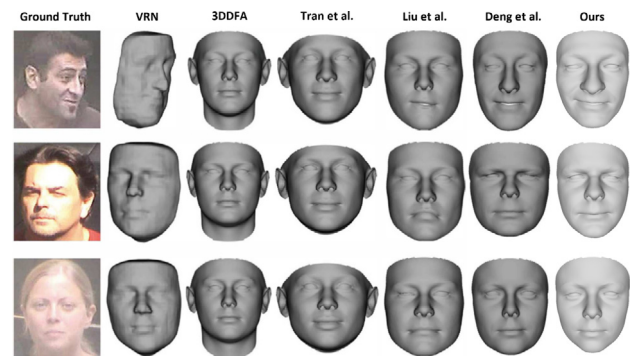


Fig. 7. Qualitative comparison of the MICC dataset. Our reconstructed shape is compared with previous works. Source: The results are taken from [Deng, Yang, et al. \(2019\)](#).

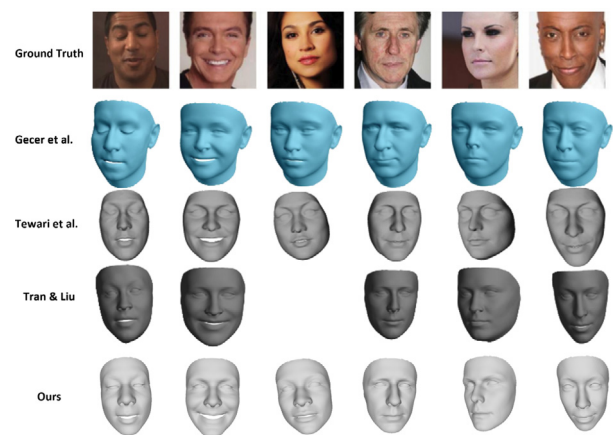


Fig. 8. Qualitative comparison of the generated face shape with previous works. Source: The results of the previous works are taken from GANFIT ([Gecer et al., 2019](#)).

works VRN ([Jackson, Bulat, Argyriou, & Tzimiropoulos, 2017](#)), 3DDFA ([Deng, Yang, et al., 2019](#); [Liu et al., 2017](#); [Tuan Tran et al., 2017](#); [Zhu et al., 2016](#)) and present the learned shapes in [Fig. 7](#). Results of the previous works are taken from ([Deng, Yang,](#)

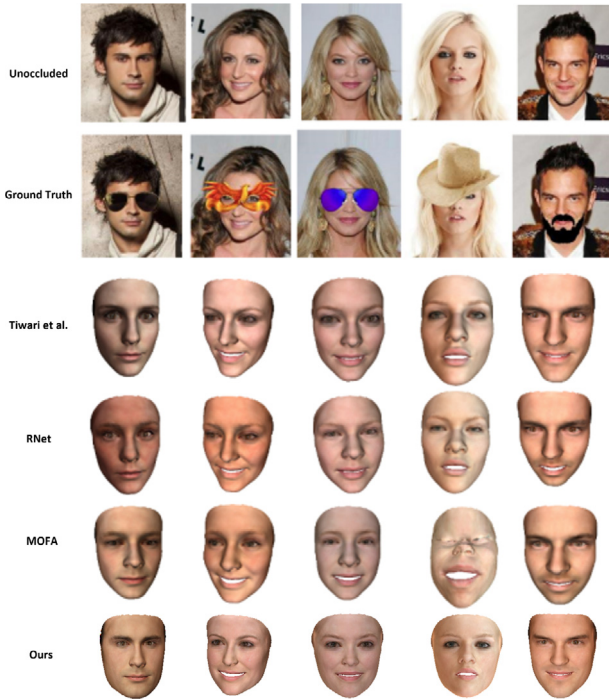


Fig. 9. Qualitative comparison with shape and texture on occluded images as ground truth. Source: The previous work outputs are taken from Tiwari, Kurmi, Venkatesh, and Chen (2022).

et al., 2019). From the visual inspection, it can be seen that our method is able to learn the shapes better than the previous works, especially near the expressive areas like the eyebrows, lips, and upper section of mouths.

Comparison of Occluded Cases: We also test our method against the face images with occlusions. Here we take sample images from a recent work (Tiwari et al., 2022), which particularly deals with face reconstruction in occlusion. The results are shown in Fig. 9. The first and second rows show the original images and the occluded images respectively. The consecutive rows show the results from the works of Tiwari et al. (2022), RNet (Deng, Yang, et al., 2019), MOFA (Tewari et al., 2017) and ours. From the results, we can see that our network learns better texture than the previous works in the occluded regions. Particularly texture-wise our result is able to preserve the identity information better than the previous works.

Comparison with other works: We also conduct a qualitative comparison with Richardson et al. (2016), Tewari et al. (2017), Tran and Liu (2018), Deng, Yang, et al. (2019) and Guo et al. (2021). The results produced by our method are comparatively better than most of the previous methods in terms of learned texture and shapes (see Fig. 6).

5.6. Subjective evaluation on face reconstruction

In this section we conducted a subjective analysis of the 3D face reconstruction results and compared our work with other related works.

5.6.1. Participants, protocols, and hypothesis

Fifteen participants (6 female and 9 male) volunteered to take part in the experiments with a median age of 27 and mean age of 26.33. All participants reported medium to high familiarity with computer graphics and digital media and are recruited through general solicitations.

Before the experiments participants are given access to a shared drive containing all the results and an excel sheet containing the detailed questionnaire with the response options. We have used a Likert scale from 1 to 5 as a level of agreement (1 – Strongly disagree, 2 – Disagree, 3 – Neither agree nor disagree, 4 – Agree, 5 – Strongly agree).

For each and every method participants are asked to give their response on the four hypotheses -

- The reconstructed face looks realistic when compared with the ground truth (realism).
- The texture of the face is well reconstructed (Texture reconstruction).
- The shape of the face is well reconstructed (Shape reconstruction).
- The overall face is well reconstructed under occlusion (Occlusion resistant).

5.6.2. Results

We computed the response of the participants and calculated the mean Opinion Score (MOS). Table 3 shows the calculated MOS and standard deviation for different methods in four different categories (hypothesis). We have compared the realism and texture constructions with Deng, Yang, et al. (2019), Gecer et al. (2019), Genova et al. (2018), Tewari et al. (2018), Tiwari et al. (2022), Tran and Liu (2018) and shape reconstruction with Deng, Yang, et al. (2019), Gecer et al. (2019), Tewari et al. (2018), Tran and Liu (2018) and the occlusion resistance with Deng, Yang, et al. (2019), Tewari et al. (2018), Tiwari et al. (2022). From the result, we can find that our method gives the best result in realism and the occlusion resistance section and came second best in texture and shape reconstruction. Though as a generic comment from the participants it has been found that it is difficult to judge the shape reconstruction results by seeing the rendered results only. Fig. 10 shows the whisker box plot for the responses to the four different hypotheses.

6. Ablation study

In this section, we validate the effectiveness of the proposed network and aggregated loss functions in the face reconstruction and texture generation task. We conduct detailed ablation experiments on the MICC Florence dataset on the following task: (1) The Feature Aggregation Module, (2) The SWIN transformer backbones, and (3) The multi-loss function. We presented the results in Mean Error (lower is better) and Standard Deviation, and the model size and complexity in terms of a number of parameters (#Params) and FLOP Counts (GFLOPs) respectively in Table 4.

6.1. Feature aggregation module

We run the experiments with and without the FAM module and analyze the results, which are shown in Table 4 (Column 'Fusion' – 'Y' denotes with and 'N' denotes without the FAM module). We can see that for all the SWIN transformer variants the network produces a lower mean error in the MICC evaluation when using the feature aggregation block. As expected the GFLOPs and the number of parameters has increased slightly after introducing the FAM module.

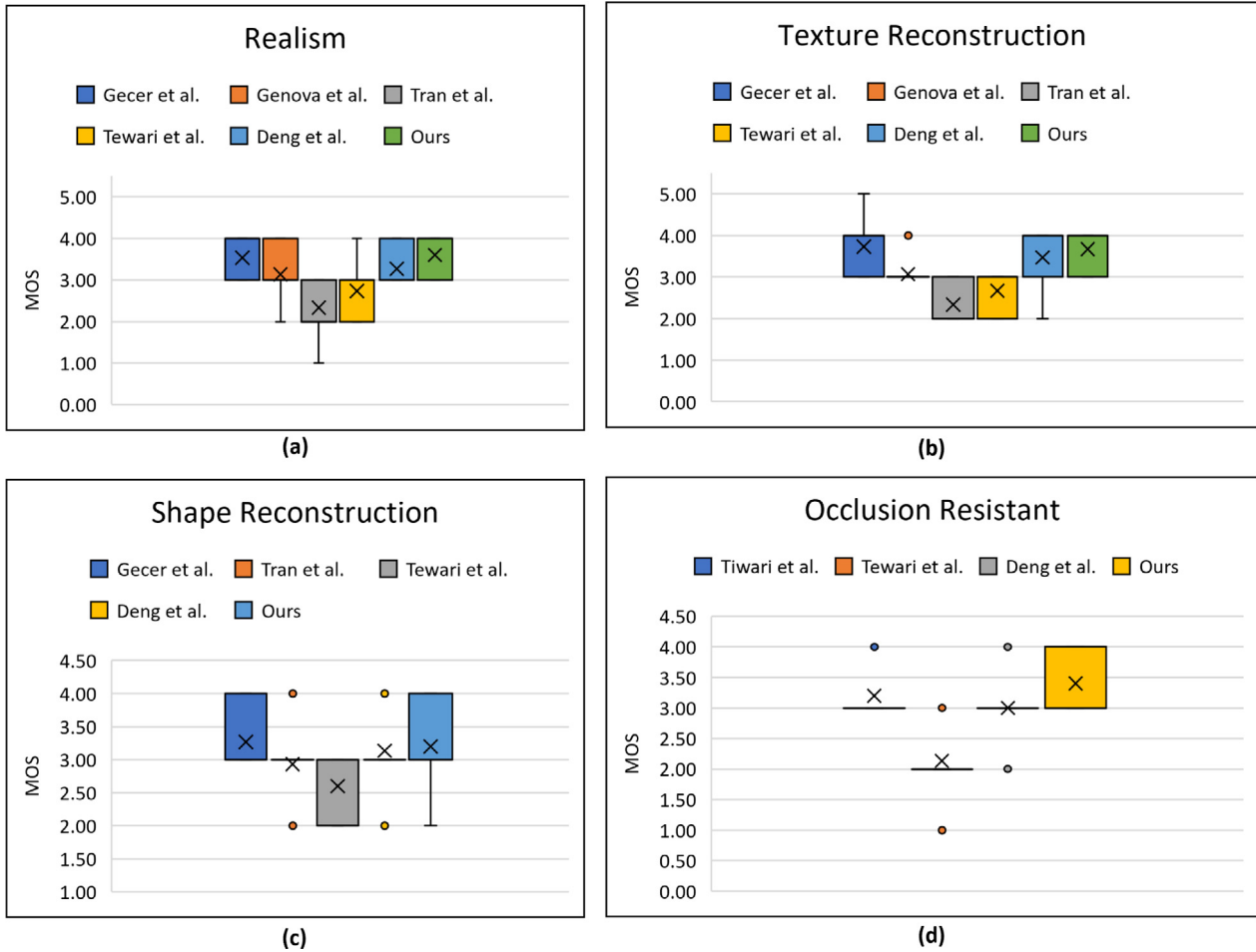


Fig. 10. Comparative results of the subjective study through whisker plots for four different hypotheses – Realism, Texture reconstruction, Shape reconstruction, and Occlusion Resistance .

Table 4

Ablation results of the network varying the Fusion method, Swin model backbone, and pretrained (trained in ImageNet data) weight variations.

*ID	Backbone	Pretrain	Fusion	#Params	GFLOPs	Co-operative		Indoor		Outdoor	
						Mean	Std.	Mean	Std.	Mean	Std.
1	Swin-T	1K	N	27.4M	6.42	1.123	0.35	1.224	0.78	1.286	0.8
2	Swin-T	1K	Y	28M	6.46	1.115	0.29	1.204	0.18	1.166	0.32
3	Swin-S	1K	N	55.1M	10.8	0.988	0.6	1.196	0.25	1.09	0.87
4	Swin-S	1K	Y	55.8M	11.1	0.983	0.45	1.18	0.29	1.02	0.46
5	Swin-B	1K	N	87M	16.8	0.978	0.56	1.12	0.74	0.99	0.35
6	Swin-B	1K	Y	87.6M	17.2	0.972	0.62	1.096	0.22	0.982	0.16
7	Swin-B	22K	N	87M	16.8	0.978	0.62	1.098	0.31	0.988	0.42
8	Swin-B	22K	Y	87.6M	17.2	0.962	0.57	1.088	0.64	0.972	0.41
9	Swin-L	1K	N	197.4M	35.6	0.968	0.44	1.098	0.55	0.97	0.12
10	Swin-L	1K	Y	198M	35.9	0.964	0.24	1.094	0.34	0.972	0.82
11	Swin-L	22K	N	197.4M	35.6	0.962	0.54	1.112	0.41	0.975	0.67
12	Swin-L	22K	Y	198M	35.9	0.956	0.23	1.086	0.24	0.964	0.22

6.2. Variation of SWIN transformer blocks

In this set of experiments, the performance of the network is chosen based on the model size. We repeat our experiments by varying the STB blocks in the regressor modules. We test with all the variants of the SWIN transformers – Tiny (Swin-T), Small (Swin-S), Base (Swin-B), and Large (Swin-L). For Swin-B and Swin-L we also test the two variants using the pre-trained models trained on the ImageNet 1K and the ImageNet 22K datasets (Column ‘Pretrain’). The remaining Swin-T and Swin-S models are trained on the Imagenet 1K dataset.

As expected the network provides in all scenarios better results (i.e., a lower mean error) when the 22K pre-trained version is used compared to the 1K pre-trained one. The best result is achieved by the Swin-L 22K version. As the model size increases from Swin-T to Swin-L, the network performance also increases with the cost of a larger GFLOPs size. From the result on rows 8 and row 12, we can see that increasing the size of the backbone from Base to Large only contributes to a limited improvement of the performance, while paying a significant increase in the computational cost. Also, the Swin-B 22k with feature fusion version outperforms the other Swin-L versions except for the Swin-L 22k with fusion. Overall the outcome of the experiments

Table 5
Ablation results of the network varying the backbones (all are pre-trained with ImageNet 22K) with image frame size 256X256.

ID	Backbone	#Params	GFLOPs	Co-operative		Indoor		Outdoor	
				Mean	Std.	Mean	Std.	Mean	Std.
1	ResNet-101	42M	8.6	1.112	0.11	1.216	0.32	1.226	0.82
2	ResNet-152	62M	11.4	1.105	0.31	1.211	0.18	1.166	0.32
3	R-101 × 3	368M	204.6	0.986	0.32	1.108	0.28	1.062	0.36
4	EffNetV2-L	105M	53.0	0.992	0.61	1.113	0.44	1.08	0.71
5	EffNetV2-XL	198M	94.0	0.988	0.25	1.042	0.12	1.02	0.82
6	ViT-B/16	82M	55.5	1.02	0.76	1.105	0.47	1.1	0.15
7	ViT-L/16	298M	191.1	0.993	0.68	1.092	0.23	1.054	0.55
8	Swin-B	87M	16.8	0.978	0.62	1.098	0.31	0.988	0.42
9	Swin-B+FAM	87.6M	17.2	0.962	0.57	1.088	0.64	0.972	0.41

shows the trade-off between the performance metrics and the computational cost (see Table 4).

6.3. Variation of backbones

To study the effect different backbones on the face reconstruction task we have repeated our experiments with the popular convolution backbones (see Table 5) like ResNet-101, ResNet-152 (He, Zhang, Ren, & Sun, 2016), R-101x3 (Kolesnikov et al., 2020), EfficientNetV2-L and XL (Tan & Le, 2021) and the variations of vanilla vision transformers ViT-B/16 and ViT-L/16 (Dosovitskiy et al., 2020). As expected the deeper variation of ResNet-101x3 performs best among the ResNets with the compromise of its huge number of parameters and FLOPs. We have also tested with the two variations of EfficientNetV2 and vanilla vision transformers (ViTs). We have found interesting observations where the EfficientNet performs better than the vanilla ViTs. Whereas the SWIN-Base backbone was able to outperform all these backbones with a comparatively smaller number of parameters and GFLOPs. Lastly, the SWIN-Base with the Feature Aggregation Module (FAM) was able to further improve the performance of the network by reducing the mean error to the lowest among all of the variations.

6.4. Multi loss functions

To study the effect of different losses on the training we also conduct several experiments to set the values of weights w_1 , w_2 , w_3 , w_4 , w_5 corresponding to the landmark loss, photometric loss, perceptual loss, shape loss, and adversarial loss respectively. We start our ablation study by observing the effect of perceptual loss and the combination of landmark loss and photometric loss individually. Following the previous works (Deng, Yang, et al., 2019; Gao et al., 2020) we put a comparatively small weight on the landmark loss compared to the other losses by setting it to 0.001 for all our experiments, as the landmark loss mostly helps to align the predicted face with the ground truth face.

From rows 1,2 and 3 in Table 6 we can see that the combination of photometric, landmark, and perceptual loss improves the result compared to using these losses individually. Combining these with the supervised shape loss and adversarial loss improves the result further. In the final experiment, we put a comparatively large weight on the photometric and the shape loss, as those two play a significant role in learning the shapes and textures. The adversarial loss helps to make the predicted shape distribution closer to the real distributions (3DMM parameters) from the 300W-LP dataset, thus making the shapes more realistic.

7. Discussion

In this work, we present a deep learning-based method that learns the 3D face model from a 2D face image with the help

of a hierarchical transformer and coarse to fine feature aggregation. Although the experimental results show that our proposed method is effective and is able to provide similar to current SOTA results it has limitations and has further scope for improvement.

- As we use a Hierarchical Vision Transformer as the feature extractor, though it has achieved a comparable result when compared to the other SOTA methods, the underlying computational cost hinders its use and deployment in edge devices. Here we can use different width and depth pruning methods to find and remove unimportant units in the transformer network to reduce its model size and computational costs.
- As we can see in Fig. 5 our method sometimes produces inaccurate shapes in the occluded regions. Here we can use different face segmentation methods to exclude these parts in the loss during backpropagation to improve its performance.
- For the supervised learning we use the 3DMM shape parameters from the 300W-LP dataset, which mostly encodes global facial deformations, therefore our method fails to recover low-dimensional facial details like wrinkles. Also, transformer-based approaches are mostly very effective in modeling non-local interactions, whereas graph convolution networks are very good at predicting neighborhood vertex interactions. Here we can introduce a graph convolution network into the transformer architecture to recover the fine-grain details.

8. Conclusion

In this work, we have explored the potential of the transformer network in the face reconstruction task. We have proposed a hierarchical transformer to extract the deep features from the face image. We have adopted the feature pyramid approach and aggregated the multi-scale features in different stages from coarse to fine. This helps to learn both local and global features from the face images. We trained the network with a hybrid loss function in a semi-supervised way without any ground truth face scans. Both qualitative and quantitative evaluations on 3D face reconstruction and 3D dense face alignment tasks demonstrate the effectiveness of our approach and ability to outperform the current SOTA task in some instances. In the subjective evaluation experiments as well our work gives better results in realism and occlusion-resistant scenarios. We also conducted an extensive set of experiments to measure the performance of the different types of Swin transformers, different feature extraction backbones, feature aggregation, and loss functions and presented their results which further provide a trade-off between model complexity, computational cost, and network performance.

Table 6

Ablation results for the network varying the weights w_1, w_2, w_3, w_4, w_5 associated to the five loss functions landmark loss (L_{lmk}), photometric loss (L_{pht}), perceptual loss (L_{perc}), shape loss (L_{shape}), and adversarial loss (L_{adv}).

w_1	w_2	w_3	w_4	w_5	Co-operative		Indoor		Outdoor	
					Mean	Std.	Mean	Std.	Mean	Std.
0.000	0.00	1.00	0.00	0.00	2.012	0.31	1.914	0.45	1.957	0.54
0.001	1.00	0.00	0.00	0.00	1.915	0.39	1.820	0.15	1.736	0.62
0.001	1.00	1.00	0.00	0.00	1.684	0.92	1.792	0.25	1.592	0.67
0.001	1.00	1.00	1.00	0.00	1.503	0.31	1.656	0.22	1.413	0.32
0.001	1.00	0.70	1.00	0.00	1.326	0.63	1.581	0.82	1.324	0.64
0.001	1.20	0.70	1.00	0.10	1.161	0.45	1.486	0.57	1.20	0.83
0.001	1.20	0.50	1.20	0.10	0.983	0.32	1.422	0.14	1.182	0.21
0.001	1.50	0.50	1.20	0.10	0.972	0.62	1.296	0.22	0.982	0.16
0.001	1.70	0.30	1.20	0.10	0.956	0.23	1.086	0.24	0.964	0.22

CRedit authorship contribution statement

Shubhajit Basak: Analysis, Ideation, Experiments, First draft.
Peter Corcoran: Review and editing the draft, Ideation, Investigation.
Rachel McDonnell: Review and editing the draft, Guidance, Ideation.
Michael Schukat: Supervision, Project administration, Draft finalization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Shubhajit Basak reports financial support was provided by Science Foundation Ireland. Funding details : Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant 18/CRT/6224

Data availability

Data will be made available on request.

References

- Bagdanov, A. D., Del Bimbo, A., & Masi, I. (2011). The florence 2D/3D hybrid face dataset. In *Proceedings of the 2011 Joint ACM workshop on human gesture and behavior understanding* (pp. 79–80).
- Bejaoui, H., Ghazouani, H., & Barhoumi, W. (2017). Fully automated facial expression recognition using 3D morphable model and mesh-local binary pattern. In *International conference on advanced concepts for intelligent vision systems* (pp. 39–50). Springer.
- Besl, P. J., & McKay, N. D. (1992). Method for registration of 3-D shapes. In *Sensor fusion IV: Control paradigms and data structures, vol. 1611* (pp. 586–606). Spie.
- Bhagavatula, C., Zhu, C., Luu, K., & Savvides, M. (2017). Faster than real-time facial alignment: A 3D spatial transformer network approach in unconstrained poses. In *Proceedings of the IEEE international conference on computer vision* (pp. 3980–3989).
- Blanz, V., Basso, C., Poggio, T., & Vetter, T. (2003). Reanimating faces in images and video. In *Computer graphics forum, vol. 22, no. 3* (pp. 641–650). Wiley Online Library.
- Blanz, V., & Vetter, T. (1999). A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on computer graphics and interactive techniques* (pp. 187–194).
- Booth, J., Antonakos, E., Ploumpis, S., Trigeorgis, G., Panagakis, Y., & Zafeiriou, S. (2017). 3D face morphable models" in-the-wild". In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 48–57).
- Booth, J., Roussos, A., Zafeiriou, S., Ponniah, A., & Dunaway, D. (2016). A 3D morphable model learnt from 10,000 faces. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5543–5552).
- Bulat, A., & Tzimiropoulos, G. (2017). How far are we from solving the 2D & 3D face alignment problem?(and a dataset of 230,000 3D facial landmarks). In *Proceedings of the IEEE international conference on computer vision* (pp. 1021–1030).
- Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018). Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition* (pp. 67–74). IEEE.
- Cao, C., Weng, Y., Zhou, S., Tong, Y., & Zhou, K. (2013). Facewarehouse: A 3D facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3), 413–425.

- Chen, C.-F. R., Fan, Q., & Panda, R. (2021). Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 357–366).
- Cudeiro, D., Bolkart, T., Laidlaw, C., Ranjan, A., & Black, M. J. (2019). Capture, learning, and synthesis of 3D speaking styles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10101–10111).
- Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4690–4699).
- Deng, Y., Yang, J., Xu, S., Chen, D., Jia, Y., & Tong, X. (2019). Accurate 3D face reconstruction with weakly-supervised learning: From single image to image set. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
- Echeagaray-Patron, B., Kober, V., Karnaukhov, V., & Kuznetsov, V. (2017). A method of face recognition using 3D facial surfaces. *Journal of Communications Technology and Electronics*, 62(6), 648–652.
- egger, B., Smith, W. A., Tewari, A., Wuhler, S., Zollhoefer, M., Beeler, T., et al. (2020). 3D morphable face models—past, present, and future. *ACM Transactions on Graphics*, 39(5), 1–38.
- Feng, Y., Wu, F., Shao, X., Wang, Y., & Zhou, X. (2018). Joint 3D face reconstruction and dense alignment with position map regression network. In *Proceedings of the European conference on computer vision* (pp. 534–551).
- Fried, O., Shechtman, E., Goldman, D. B., & Finkelstein, A. (2016). Perspective-aware manipulation of portrait photos. *ACM Transactions on Graphics*, 35(4), 1–10.
- Gao, Z., Zhang, J., Guo, Y., Ma, C., Zhai, G., & Yang, X. (2020). Semi-supervised 3D face representation learning from unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 348–349).
- Garrido, P., Valgaerts, L., Sarmadi, H., Steiner, I., Varanasi, K., Perez, P., et al. (2015). VDUB: Modifying face video of actors for plausible visual alignment to a dubbed audio track. In *Computer graphics forum, vol. 34, no. 2* (pp. 193–204). Wiley Online Library.
- Gecer, B., Ploumpis, S., Kotsia, I., & Zafeiriou, S. (2019). Ganfit: Generative adversarial network fitting for high fidelity 3D face reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1155–1164).
- Geng, Z., Cao, C., & Tulyakov, S. (2019). 3D guided fine-grained face manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9821–9830).
- Genova, K., Cole, F., Maschinot, A., Sarna, A., Vlastic, D., & Freeman, W. T. (2018). Unsupervised training for 3D morphable model regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8377–8386).
- Guo, Y., Cai, J., Jiang, B., Zheng, J., et al. (2018). CNN-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(6), 1294–1307.
- Guo, Y., Cai, L., & Zhang, J. (2021). 3D face from X: Learning face shape from diverse sources. *IEEE Transactions on Image Processing*, 30, 3815–3827.
- Guo, J., Yu, J., Lattas, A., & Deng, J. (2022). Perspective reconstruction of human faces by joint mesh and landmark regression. arXiv preprint arXiv:2208.07142.
- Guo, Y., Zhang, L., Hu, Y., He, X., & Gao, J. (2016). Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision* (pp. 87–102). Springer.
- Guo, J., Zhu, X., Yang, Y., Yang, F., Lei, Z., & Li, S. Z. (2020). Towards fast, accurate and stable 3D dense face alignment. In *European conference on computer vision* (pp. 152–168). Springer.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

- Huber, P., Hu, G., Tena, R., Mortazavian, P., Koppen, P., Christmas, W. J., et al. (2016). A multiresolution 3D morphable face model and fitting framework. In *Proceedings of the 11th international joint conference on computer vision, imaging and computer graphics theory and applications*. University of Surrey.
- Jackson, A. S., Bulat, A., Argyriou, V., & Tzimiropoulos, G. (2017). Large pose 3D face reconstruction from a single image via direct volumetric CNN regression. In *Proceedings of the IEEE international conference on computer vision* (pp. 1031–1039).
- Jiang, L., Wu, X.-J., & Kittler, J. (2019). Dual attention MobDenseNet (damdnet) for robust 3D face alignment. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*.
- Karras, T., Aila, T., Laine, S., Herva, A., & Lehtinen, J. (2017). Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics*, 36(4), 1–12.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., et al. (2020). Big transfer (bit): General visual representation learning. In *European conference on computer vision* (pp. 491–507). Springer.
- Laine, S., Hellsten, J., Karras, T., Seol, Y., Lehtinen, J., & Aila, T. (2020). Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6).
- Li, T., Bolkart, T., Black, M. J., Li, H., & Romero, J. (2017). Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics*, 36(6), 194:1–194:17.
- Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117–2125).
- Liu, Y., Jourabloo, A., Ren, W., & Liu, X. (2017). Dense face alignment. In *Proceedings of the IEEE international conference on computer vision workshops* (pp. 1619–1628).
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012–10022).
- Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of international conference on computer vision*.
- Mathieu, M., Couprie, C., & LeCun, Y. (2015). Deep multi-scale video prediction beyond mean square error. arXiv preprint arXiv:1511.05440.
- Nirkin, Y., Masi, I., Tuan, A. T., Hassner, T., & Medioni, G. (2018). On face segmentation, face swapping, and face perception. In *2018 13th IEEE international conference on automatic face & gesture recognition* (pp. 98–105). IEEE.
- Paysan, P., Knothe, R., Amberg, B., Romdhani, S., & Vetter, T. (2009). A 3D face model for pose and illumination invariant face recognition. In *2009 Sixth IEEE international conference on advanced video and signal based surveillance* (pp. 296–301). IEEE.
- Piao, J., Qian, C., & Li, H. (2019). Semi-supervised monocular 3D face reconstruction with end-to-end shape-preserved domain transfer. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9398–9407).
- Ramamoorthi, R., & Hanrahan, P. (2001). A signal-processing framework for inverse rendering. In *Proceedings of the 28th annual conference on computer graphics and interactive techniques* (pp. 117–128).
- Ranjan, A., Bolkart, T., Sanyal, S., & Black, M. J. (2018). Generating 3D faces using convolutional mesh autoencoders. In *Proceedings of the European conference on computer vision* (pp. 704–720).
- Richardson, E., Sela, M., & Kimmel, R. (2016). 3D face reconstruction by learning from synthetic data. In *2016 fourth international conference on 3D vision* (pp. 460–469). IEEE.
- Roth, J., Tong, Y., & Liu, X. (2016). Adaptive 3D face reconstruction from unconstrained photo collections. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4197–4206).
- Ruan, Z., Zou, C., Wu, L., Wu, G., & Wang, L. (2021). Sadrnet: Self-aligned dual face regression networks for robust 3D dense face alignment and reconstruction. *IEEE Transactions on Image Processing*, 30, 5793–5806.
- Shang, J., Shen, T., Li, S., Zhou, L., Zhen, M., Fang, T., et al. (2020). Self-supervised monocular 3D face reconstruction by occlusion-aware multi-view geometry consistency. In *European conference on computer vision* (pp. 53–70). Springer.
- Sheng, H., Cai, S., Liu, Y., Deng, B., Huang, J., Hua, X.-S., et al. (2021). Improving 3D object detection with channel-wise transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2743–2752).
- Shu, Z., Yumer, E., Hadap, S., Sunkavalli, K., Shechtman, E., & Samaras, D. (2017). Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5541–5550).
- Tan, M., & Le, Q. (2021). Efficientnetv2: Smaller models and faster training. In *International conference on machine learning* (pp. 10096–10106). PMLR.
- Tewari, A., Zollhöfer, M., Garrido, P., Bernard, F., Kim, H., Pérez, P., et al. (2018). Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2549–2559).
- Tewari, A., Zollhofer, M., Kim, H., Garrido, P., Bernard, F., Perez, P., et al. (2017). Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE international conference on computer vision workshops* (pp. 1274–1283).
- Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., & Nießner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2387–2395).
- Tiwari, H., Kurmi, V. K., Venkatesh, K., & Chen, Y.-S. (2022). Occlusion resistant network for 3D face reconstruction. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 813–822).
- Tráň, A. T., Hassner, T., Masi, I., Paz, E., Nirkin, Y., & Medioni, G. (2018). Extreme 3D face reconstruction: Seeing through occlusions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3935–3944).
- Tran, L., & Liu, X. (2018). Nonlinear 3D face morphable model. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7346–7355).
- Tran, L., & Liu, X. (2019). On learning 3D face morphable model from in-the-wild images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1), 157–171.
- Tu, X., Zhao, J., Xie, M., Jiang, Z., Balamurugan, A., Luo, Y., et al. (2020). 3D face reconstruction from a single image assisted by 2D face images in the wild. *IEEE Transactions on Multimedia*, 23, 1160–1172.
- Tuan Tran, A., Hassner, T., Masi, I., & Medioni, G. (2017). Regressing robust and discriminative 3D morphable models with a very deep neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5163–5172).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- Wu, J., Huang, Z., Thoma, J., Acharya, D., & Van Gool, L. (2018). Wasserstein divergence for GANs. In *Proceedings of the European conference on computer vision* (pp. 653–668).
- Wu, S., Ruppert, C., & Vedaldi, A. (2020). Unsupervised learning of probably symmetric deformable 3D objects from images in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1–10).
- Zhao, J., Xiong, L., Li, J., Xing, J., Yan, S., & Feng, J. (2018). 3D-aided dual-agent gans for unconstrained face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(10), 2380–2394.
- Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., et al. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6881–6890).
- Zhou, Y., Deng, J., Kotsia, I., & Zafeiriou, S. (2019). Dense 3D face decoding over 2500fps: Joint texture & shape convolutional mesh decoders. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1097–1106).
- Zhu, X., Lei, Z., Liu, X., Shi, H., & Li, S. Z. (2016). Face alignment across large poses: A 3D solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 146–155).
- Zhu, H., Sun, X., Li, Y., Ma, K., Zhou, S. K., & Zheng, Y. (2022). DFTR: Depth-supervised hierarchical feature fusion transformer for salient object detection. arXiv preprint arXiv:2203.06429.
- Zielonka, W., Bolkart, T., & Thies, J. (2022). Towards metrical reconstruction of human faces. arXiv preprint arXiv:2204.06607.

Chapter 5

Lightweight dense facial Landmark Prediction

5.1 Background

As discussed in the previous chapter, with the help of highly complex deep neural network models, we are able to recover the detailed face shape from uncalibrated face images. However, most of these methods depend on some kind of statistical priors of face shape like a 3DMM and the sparse face landmarks for face alignments. Some of the previous works also used additional signals beyond color images, like facial depth [131, 10], optical flow [27], or multi-view stereo [17, 121], and then optimized using geometric and temporal prior. Each of these methods can produce very detailed results but take a very long time to compute. At the same time, the model size and huge computational requirements make these approaches not suitable for real-time applications in edge devices. Therefore it is still a very challenging task to implement a face modeling pipeline on limited computational cost systems such as mobile or embedded systems.

To reduce the dependencies on the priors, such as the statistical models, estimating 3D landmarks on the face can work as an alternative to estimating the face structure. These landmarks work as a point of correspondence across the face. But all the publicly available datasets mostly contain a sparse set of 68 facial landmarks, which fails to encode the full face structure. So increasing the number of these landmarks can help to learn face geometry better. Unfortunately, annotating a real face with dense landmarks is highly ambiguous and expensive. Some of the previous methods, like Wood et al. [150], rely on synthetic data alone. Though the authors have detailed ground truth annotations like albedo, normals, depth, and dense landmarks, none of these data is publicly available. The authors also

proposed a method [152] to learn the dense landmarks as a Gaussian uncertainty from those synthetic data and fit a 3DMM model from those dense key points only. Some other methods [38, 49, 168] use pseudo-labels model-fitting approaches like fitting an existing 3DMM model to generate synthetic landmarks. Jeni et al. [77] predicted dense frontal face landmarks with cascade regressions. They created 1024 dense 3D landmark annotations from 3D scan datasets [165, 166] through an iterative method. While Kartynnik et al. [82] used a predefined mesh topology of 468 points arranged in fixed quads and fit a 3DMM model to a large set of in-the-wild images to create ground truth 3D dense annotations of key points. They later employed direct regression to predict these landmarks from face images. Some other methods [6, 49] used a different method to unwrap each pixel of the face as a position map and regress the position in 3D space. They created the position map by fitting the Basel Face Model (BFM) [106] from the 300WLP dataset [168], which has the 3DMM parameters associated with more than 60k in the wild images. As we don't have access to such massive 3D scan data, the same position map data can be an option to create the ground truth dense landmark.

5.2 Research Objective

As discussed in the above section, as we don't have access to large 3D scan data, generating position maps similar to Feng et al. [49] can be an alternative. The position map records the 3D shape of the complete face in UV space as a 2D representation, where each pixel value has the 3D position information of that pixel. It provides correspondence to the semantic meaning of each point on the UV space. Their method aligns a 3D face model to the corresponding 2D face image and stores the 3D position of the points. We can apply the same to extract dense key points to create the ground truth data, before using direct regression to train a model that can predict those dense landmarks in 3D space. Overall, the main objective of this study is the followings:

- Extract a dense key point mesh topology from the existing UV position map extracted by Feng et al. [49] that will have the same semantic meaning across all faces.
- Following the topology, create the ground truth data of face images and their corresponding dense key points.
- Create a regression model to perform the direct regression task.
- Evaluate the model performance in terms of 3D key points.

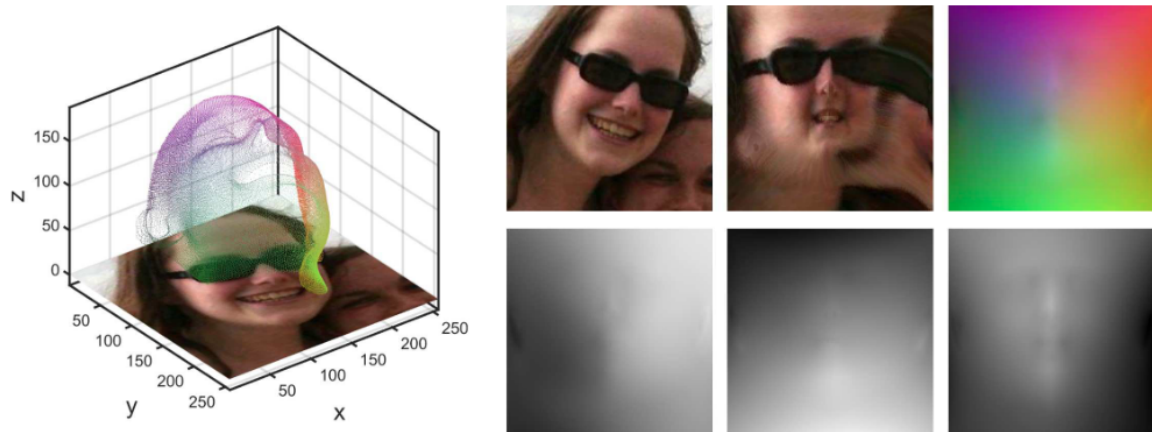


Fig. 5.1 UV position map example from [49]. Left: 3D plot of the corresponding position map on top of the 2D RGB image. Right: The first row is the 2D RGB image and the corresponding extracted texture and position map. The second row shows the x,y, and z channels of the position map data.

5.3 Summary of Contribution

To achieve the above objective, we first propose a methodology to generate the dense key points of 520 face landmarks from the position map data. Then we create a lightweight regressor network to build a model that will predict those key points from monocular face images. We will discuss these in more detail in the following section:

5.3.1 Dense Facial Landmark Data Generation from UV Map

As stated in the previous section, Feng et al. [49] proposed a 3D facial representation based on the UV position map. They used the UV space to store the 3D position points from the 3D face model aligned with the 2D facial image. They assume the projection from the 3D model on the 2D image as a weak perspective projection and define the 3D facial position as a Left-hand Cartesian coordinate system. The ground truth 3D facial shape exactly matches the 2D image when projected to the x-y plane. They define the position map as $Pos(u_i, v_i) = (x_i, y_i, z_i)$, where (u_i, v_i) represents the i th point in face surface and (x_i, y_i, z_i) represents the corresponding 3D position of facial mesh with (x_i, y_i) being the corresponding 2D position of the face in the input RGB image and z_i representing the depth value of the corresponding point. Figure 5.1 shows an example of the position map data taken from [49].

We followed the same representation and used their pipeline to build the raw data from the 300W-LP [168] dataset. This contains more than 60k unconstrained face images with fitted 3DMM parameters which are based on the Basel Face Model. They used the parameterized

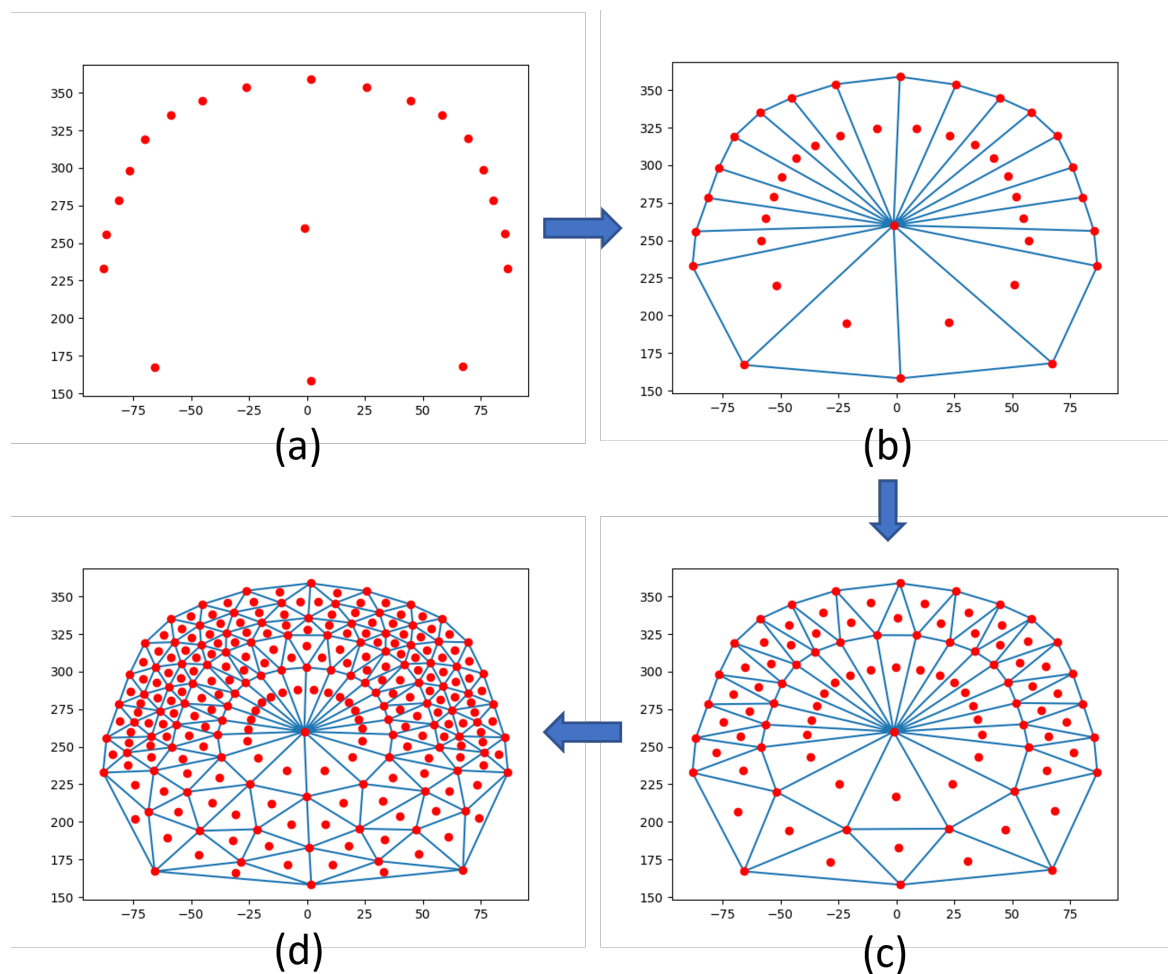


Fig. 5.2 Selection of key points through Delaunay Triangulation. (a) Initial selected key points across the jaw, forehead, and nose tip (b) First iteration of Delaunay triangulation and centroid selection (c) Second iteration of Delaunay triangulation and centroid selection (d) Third iteration of Delaunay triangulation and centroid selection

UV coordinates from Bas et al. [11], which computes a Tutte embedding [51] with conformal Laplacian weight and then maps the mesh boundary to a square. So we can filter this UV position map data to create a dense face landmark. The 3DMM face template that was used by Feng et al. [49] has a total of 43867 vertices. Out of these, we have sampled 520 vertices. To sample, we have followed the following steps as shown in figure 5.2 -

- First, we have selected 18 key points across the jaw and one key point on the nose tip from the 68 key points provided.
- Then we run the Delaunay triangulation [93] on the selected points and select the centroids of the three vertices of each triangle.

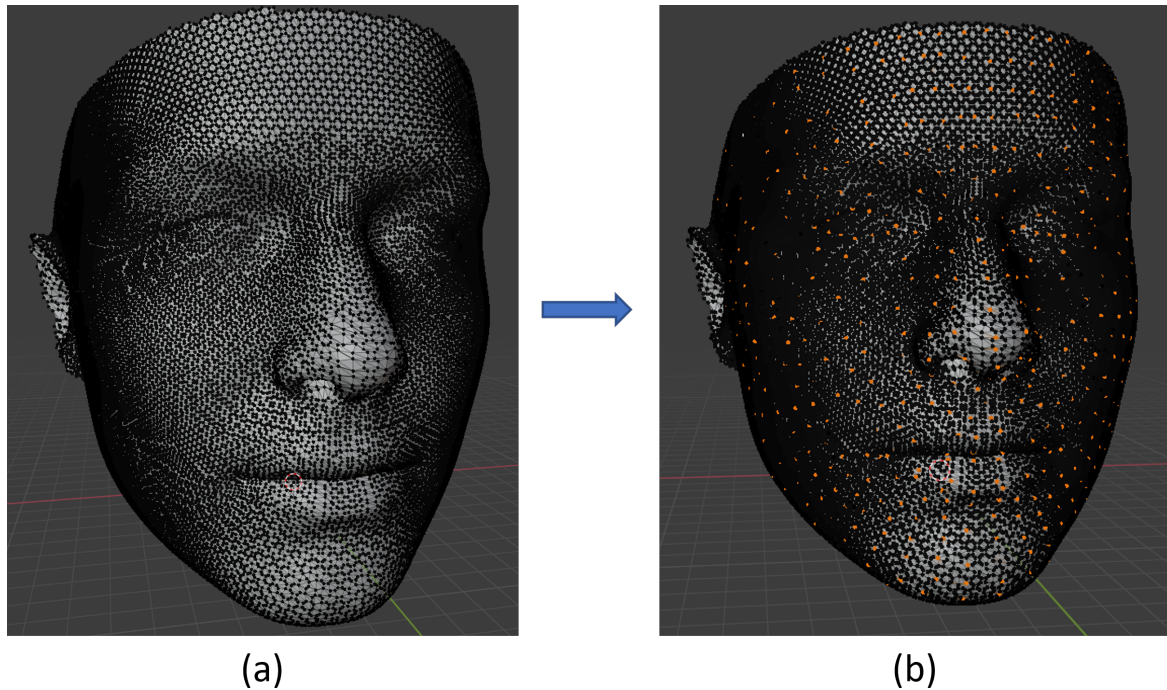


Fig. 5.3 (a) Template mesh in Blender (b) Final selected vertices highlighted on the template in Blender

- We repeat the same step another two times and have the final key points
- Finally, we select these key points across the template mesh and manually select the rest of the key points and rectify some of the already selected key points in Blender.

After these iterations, the final version of the ground truth data has the RGB face images and their corresponding 520 face key points which includes the popular 68 key points set in it. Figure 5.3 shows the final selected key points on a face mesh in Blender. Figure 5.4 shows some of the samples from the ground truth data. The whole dataset contains around 61k pairs of ground truth images and their corresponding ground truth position map data saved in numpy format. Further, we expanded the data by applying a horizontal flip which made the total dataset size to 120k of paired images and their position map data.

5.3.2 Dense Facial Landmark Prediction using Regression

As we have around 120k pairs of ground truth face images in the wild and their corresponding ground truth facial key points, we formulate the problem as a direct regression of those 520 key points from a monocular face image. We build a model with a standard feature extractor with a classifier head. The trained model will predict a continuous value of three positions

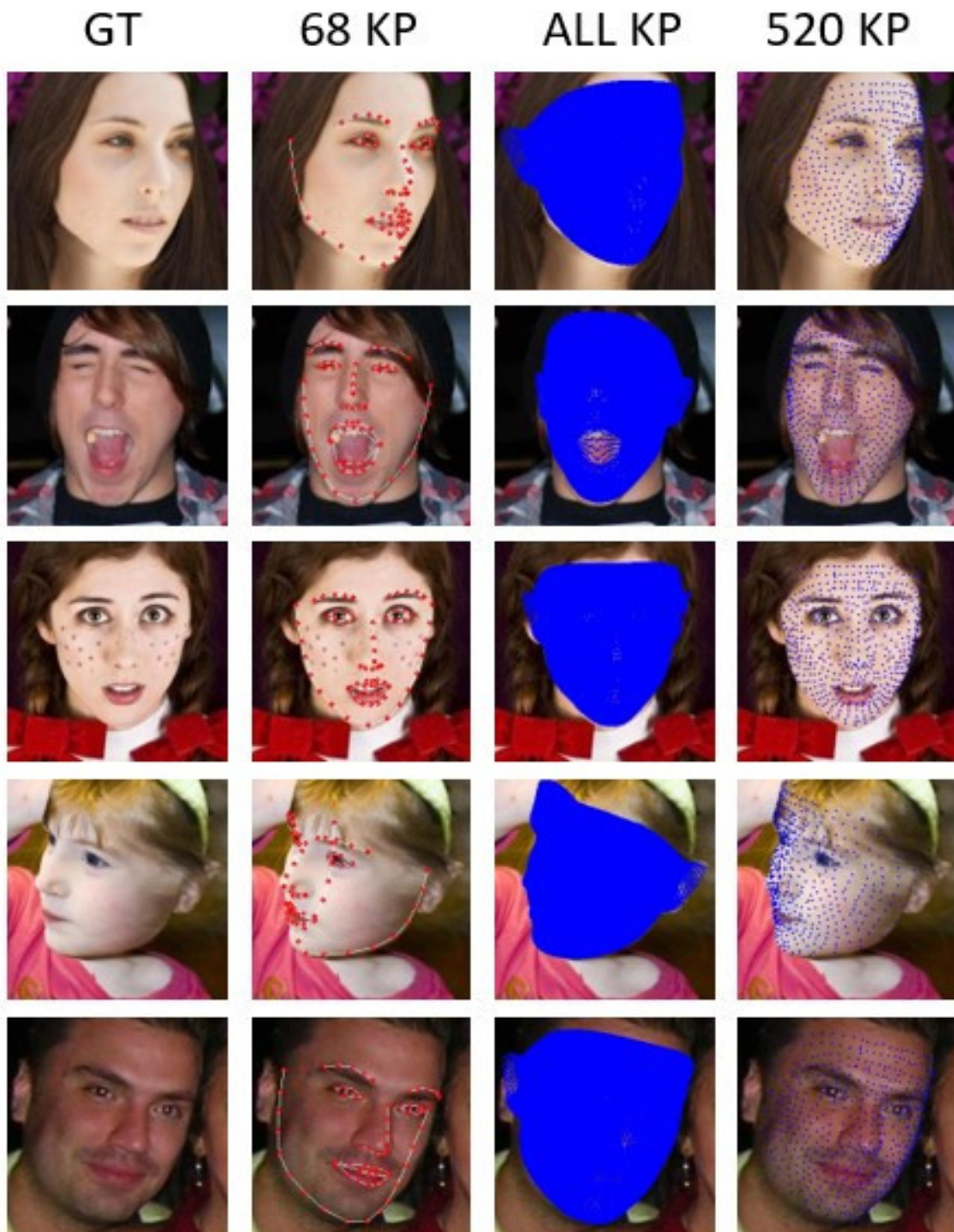


Fig. 5.4 Sample ground truth data. 1st column in the RGB image, 2nd column shows the selected 68 key points, 3rd column shows the fully selected 43867 vertices. 4th column shows the final selected 520 vertices.

(x,y,z) for those 520 3D landmarks. We choose the total number of classes as $520 \times 3 = 1560$. As the feature extractor, we have chosen two popular backbones, Resnet50 and MobilenetV2.

The standard loss function that is typically used for any landmark estimator is the $L1$, and $L2$ loss or the Mean Square Error loss [24–26, 142]. But the $L2$ loss ($L2(x) = x^2/2$) function is sensitive to outliers. So Rashid et al. [111] used *smoothL1* loss which is defined as -

$$smoothL1(x) = \begin{cases} x^2/2, & \text{if } |x| < 1 \\ |x| - 1/2, & \text{otherwise} \end{cases} \quad (5.1)$$

Both $L1$ ($L1(x) = |x|$) and *smoothL1* perform well for outliers, but they produce a very small value for small landmark differences. This hinders the network training for small errors. To solve this issue, Feng et al. [50] proposed a new loss called Wing loss which pays more attention to small and medium errors. They combined the $L1$ loss for the large landmark deviations and $\ln(\cdot)$ for small deviations as follows -

$$wing(x) = \begin{cases} w \ln(1 + |x|/\varepsilon), & \text{if } |x| < w \\ |x| - C, & \text{otherwise} \end{cases} \quad (5.2)$$

where $C = w - w \ln(1 + w/\varepsilon)$, w and ε are the hyperparameters ($w = 15$, $\varepsilon = 3$ in the paper). In this work as well we combined the Mean Square Error loss with the Wing loss to define a hybrid loss function as -

$$L = w_1 L_{Wing} + w_2 L_{MSE} \quad (5.3)$$

Through an ablation study, we set the weight of these two loss terms as $w_1 = 1.5$ and $w_2 = 0.5$.

As we don't have any evaluation or test dataset that has the 3DMM parameters or the position map data available, we evaluated our trained model on the 3D face alignment task. To measure the face alignment quantitatively, we use the normalized mean error (NME) as the evaluation metric. NME is computed as the normalized mean Euclidean distance between each set of corresponding landmarks in the predicted result l and the ground truth l' :

$$NME = \frac{1}{N} \sum_{i=1}^N \frac{\|l_i - l'_i\|_2}{d} \quad (5.4)$$

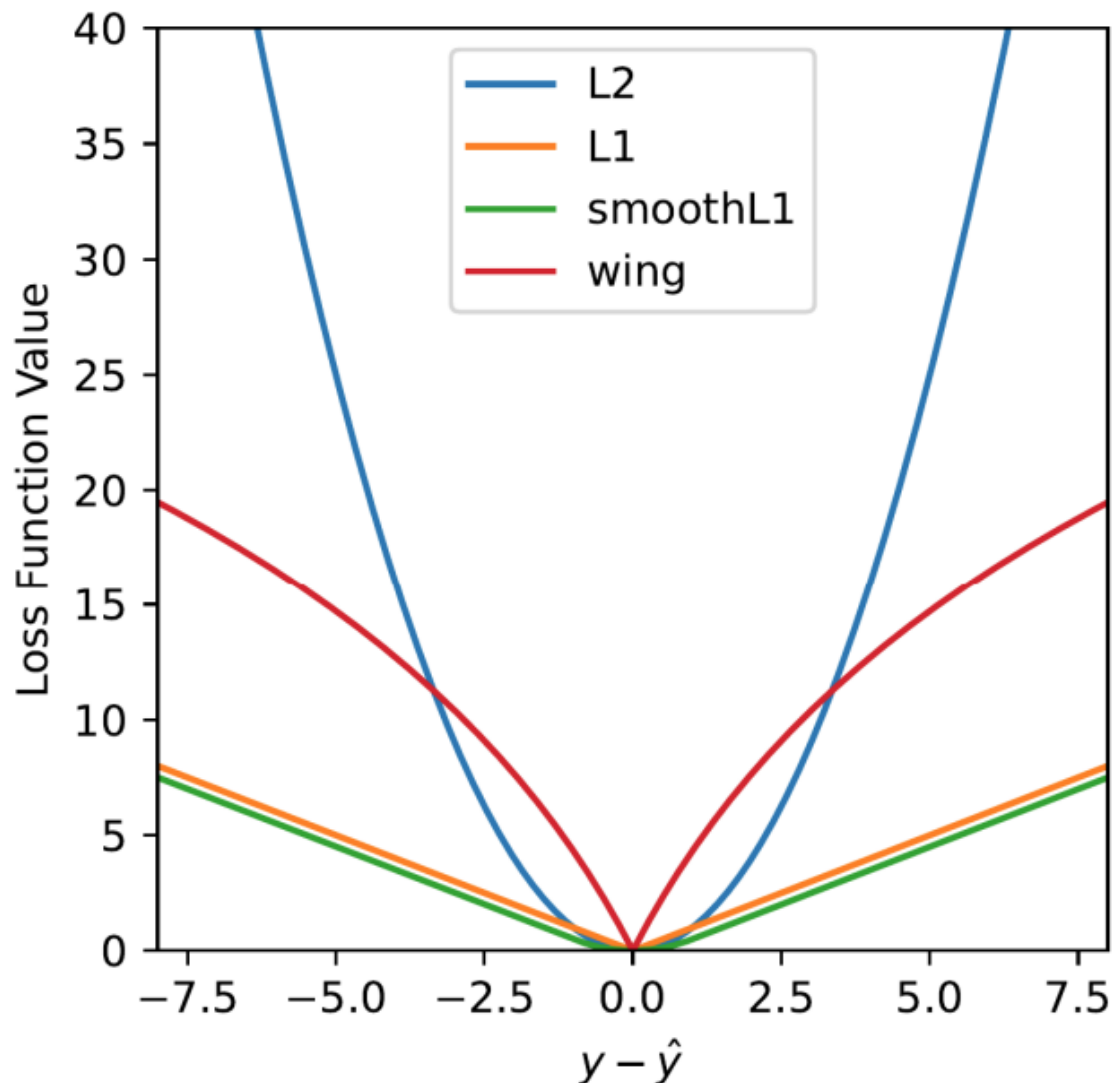


Fig. 5.5 Loss function comparison: L2, L1, smoothL1, Wing (with $w = 15$, $\varepsilon = 3$). The plot shows the loss value against the error between the ground truth and the predicted value [83]. The quadratic growth of the L2 loss makes it sensitive to the outliers, while L2, L1, and smoothL1 yield a very small value for small errors between the ground truth and the predicted values. On the contrary, Wing loss is less sensitive to outliers and is much more sensitive to medium-to-small errors, which improves the training overall.

Following the previous works [116], the normalization factor d is computed as $\sqrt{h * w}$, where h and w are the height and width of the bounding box, respectively. Similar to [49, 116] for 2D and 3D sparse alignment, we consider all 68 landmark points. We divide the dataset based on the yaw angles $(0^\circ, 30^\circ)$, $(30^\circ, 60^\circ)$ and $(60^\circ, 90^\circ)$ and a balanced subset created by taking a random sample from the whole dataset. We benchmarked our model on the

Table 5.1 Quantitative evaluation on AFLW2000-3D dataset on facial alignment task.

Method	0 to 30	30 to 60	60 to 90	All
ESR [28]	4.60	6.70	12.67	7.99
3DDFA [168]	3.43	4.24	7.17	4.94
DenseCorr [161]	3.62	6.06	9.56	6.41
3DSTN [18]	3.15	4.33	5.98	4.49
3D-FAN [26]	3.16	3.53	4.60	3.76
3DDFA TPAMI [169]	2.84	3.57	4.96	3.79
PRNet [49]	2.75	3.51	4.61	3.62
2DASL [137]	2.75	3.46	4.45	3.55
3DDFA V2[64]	2.63	3.420	4.48	3.51
Ours	2.86	3.68	4.76	3.77

Table 5.2 Quantitative evaluation on AFLW dataset with 21-point landmark definition on facial alignment task.

Method	0 to 30	30 to 60	60 to 90	All
ESR [28]	5.66	7.12	11.94	8.24
3DDFA [168]	4.75	4.83	6.39	5.32
3D-FAN [26]	4.40	4.52	5.17	4.69
3DSTN [18]	3.55	3.92	5.21	4.23
3DDFA TPAMI [169]	4.11	4.38	5.16	4.55
PRNet [49]	4.19	4.69	5.45	4.77
3DDFA V2[64]	3.98	4.31	4.99	4.43
Ours	4.04	4.45	5.2	4.57

Table 5.3 Comparative analysis with two different backbones Mobilenet-V2 and Resnet-18 of Quantitative result on AFLW-3D dataset on facial alignment task and the computational requirement.

Backbone	0 to 30	30 to 60	60 to 90	All	gMac	gFlop	# Params
Resnet-18	2.88	3.72	4.82	3.83	5.13	2.56	16.03M
Mobilenet-V2	2.86	3.68	4.76	3.77	0.39	0.19	4.18M

widely used AFLW2000-3D dataset. It is an in-the-wild face dataset with a large variation in illumination, pose, occlusion, and expression. It has 2000 images with 68 3D face landmark points for face alignment.

Following 3DDFA-V2 [64], we have also evaluated our work using the AFLW full set (21K test images with 21-point landmarks). We followed the same split and showed the results for different angles in table 5.2.

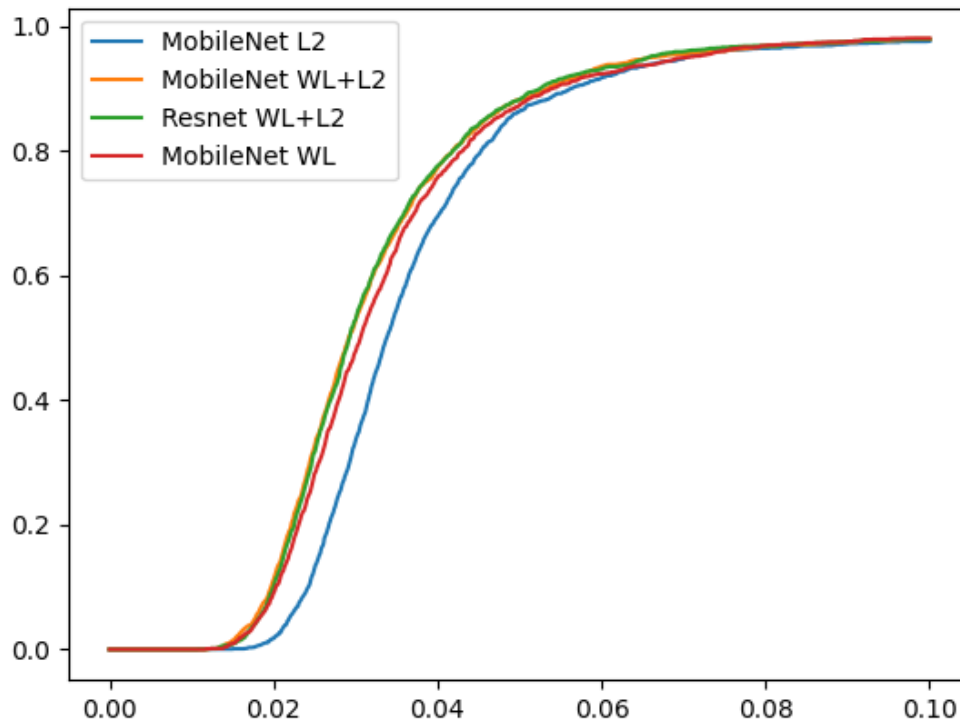


Fig. 5.6 Cumulative Errors (NME) Distribution (CED) curves on AFLW2000-3D. Evaluation is performed on 68 landmarks with coordinates. Overall 2000 images from the AFLW2000-3D dataset are used. The backbone and loss functions are also shown in the legend. WL stands for Wing Loss, and L2 stands for MSE loss

5.4 Discussion on Contribution

As there is no public data set available for dense landmarks, we have proposed a pipeline to create ground truth data for 520 key points. With the help of that data generated, we have trained a key point detection network with two popular backbones, Resnet18 and MobileNetV2. As we don't have access to any evaluation dataset which has dense landmarks, we evaluated our model on a 3D face alignment task for 68 key points. We have used a hybrid loss function for the learning, which is a combination of MSE and Wing loss. Experimental results show that with the help of a hybrid loss function, we are able to achieve near SOTA performance on both AFLW2000-3D and AFLW benchmarks. Also, the MobilenetV2-based model is comparatively lightweight and requires fewer computational resources. Table 5.3 shows a comparative analysis of the Resnet and Mobilenet-based networks in terms of their computational resource requirement. We have also conducted an ablation study on the effect

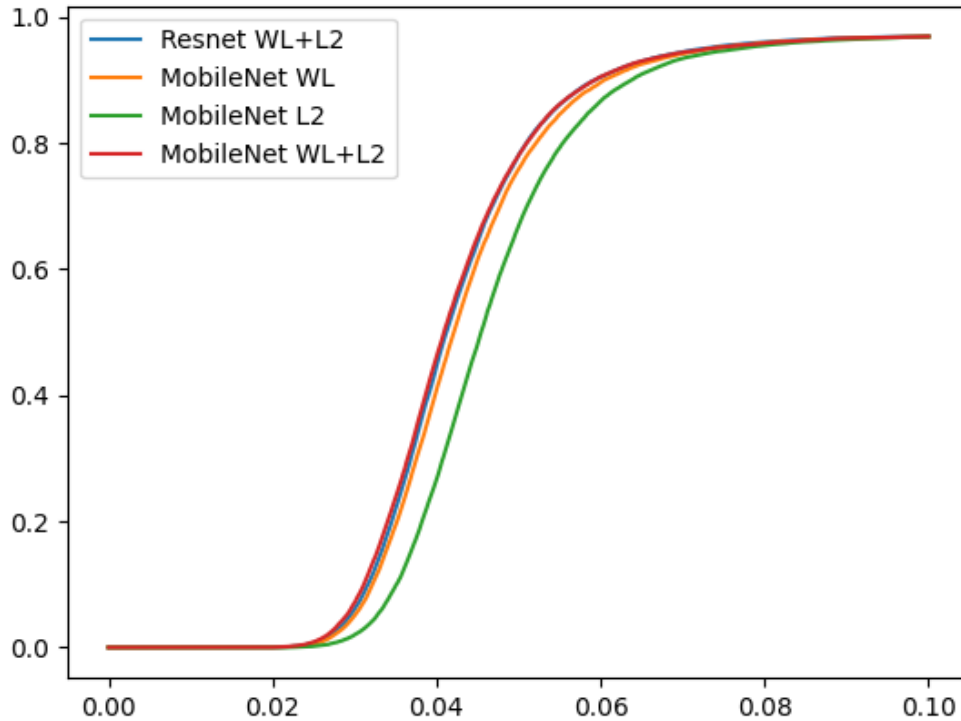


Fig. 5.7 Cumulative Errors (NME) Distribution (CED) curves on AFLW with 21 point landmarks. Overall 21k images from the AFLW dataset are used here. The backbone and loss functions are also shown in the legend. WL stands for Wing Loss, and L2 stands for MSE loss.

of the hybrid loss function. Figure 5.6 and 5.7 shows the cumulative error distribution curves based on NME for the AFLW-3D and AFLW dataset. In both cases, a combination of Wing Loss and MSE performs better than the rest.

Though by visual inspection, the results of the model look good, due to the lack of ground truth test data we are only able to evaluate the model against the 3D facial alignment task. In the future, we can extend this work and use those 520 key points to fit an existing statistical (e.g., 3DMM) model to the face and evaluate the full face reconstruction benchmark.

Chapter 6

Additional Contributions

In this chapter, some of my secondary publications are briefly mentioned.

6.1 Speech-driven Video Editing via Audio-Conditioned Diffusion Model

6.1.1 Background

One of the most popular facial analysis and modification tasks is visual dubbing. This task synthesizes a lip-synced talking head video by inputting the corresponding audio and mesh vertex, facial image, or video. The model first learns from the visual appearance data and then dynamically maps the lower-dimensional speech signal to the high-dimensional video signal data such as facial expression, facial action, and human lip shape. Finally, the learned network performs the video rendering and outputs the multi-modal video data. It has several real-world applications, like translating a video into a different language or modifying the speech after recording it.

Before the deep learning era, researchers mainly adopted cross-modal retrieval methods [56, 55, 132, 23] and Hidden Markov Model (HMM) [155] to accomplish talking head generation tasks. With the rapid development of deep learning, it has become the de-facto method for this task. The deep-learning-based tasks can be broadly divided into two categories - pipeline-based methods and end-to-end methods. The pipeline-based techniques mainly consist of two steps: first, the low dimensional driving source data, such as the audio signals, are mapped to the facial parameters like face landmarks and facial coefficients (e.g., 3DMM parameters). Then the facial parameters are passed to a video rendering module to generate the faces by GPU rendering, video editing, or GAN-based generative models.

The earliest pioneering landmark-based method was proposed by Suwajanakorn et al. [128]. They used a single-layer LSTM to map the low-dimensional speech signal into nonlinear lip keys and passed them to face texture synthesis and video construction. Like them, Kumar et al. [92] used an LSTM+UNet-based architecture and Pix2Pix-based image synthesis method. But both these methods are limited to a single training speaker as they trained on an Obama-speaking video dataset. To overcome this limitation, Jalalifar et al. [76] introduced a basic conditional generative adversarial network (C-GAN) as a stand-alone network for audio-to-video mapping to generate video from the learned landmarks. It can learn from any audio as a driving source, thus minimizing the dependency on a person's specific audio source. Kim et al. [88] introduced the 3DMM [20] as the facial parameter learning phase. As 3DMM is a full-face parametric model, this method covered full control over the facial action parameters, like expressions, shape, and scene illuminations. Though the method does not consider temporal coherence, the lips in the consecutive frames are not aligned properly. However, most models are trained on particular speaker-specific data and cannot generalize among different identities. To overcome this issue, Cudeiro et al. [35] proposed a model called VOCA, which fused the audio features extracted by DeepSpeech from different speakers and output the displacement data of the 3D vertices from the FLAME parametric model.

Although these pipeline methods were very popular in the early deep-learning era, there are major drawbacks. These methods have a complex pipeline of processing, time-consuming and expensive facial parameter annotations, and depend on auxiliary techniques like facial landmark detection and 3D reconstructions. To overcome this researchers began recently to study the end-to-end approach for talking head video synthesis, where the goal is to generate the lip-synced face videos from the driving source (like audio) without involving any facial parameters like key points as an intermediate learning step. One of the earliest methods to explore this end-to-end strategy is the Speech2Vid [34], which consists of an audio encoder, an speaker identity encoder, an image decoder, and a deblurring module. The image decoder takes the audio feature vector and identity feature vector. It performs a feature fusion through a transposed convolution and an up-sampling method to synthesize an output image. But, it does not consider the continuity in time series and produces incoherent video sequences, skipped frames, and jitters. To overcome this, researchers proposed GAN-based methods and introduced more efficient learning objectives. Vougioukas et al. [139] first introduced the GAN-based speech-driven video generation. They proposed an end-to-end approach to generate talking head videos using a single image of a person and an audio clip of the speech without relying on any hand-crafted intermediate features. In recent work, Yin et al.

proposed StyleHEAT [159], which utilizes StyleGAN [81] to synthesize talking faces guided by speech embeddings.

6.1.2 Research Objective

Despite the popularity of GAN in image generation, their application in speech-driven video synthesis is limited by some drawbacks. One of the foremost reasons is the difficulty of gaining stability in GAN training. It often requires extensive architectural search and tuning of model parameters to achieve convergence. The stability of the training can be improved by using additional guidance like face masks or driving frames. This limits the facial reenactment task like talking head generation and reduces the ability to generate original head movements and expressions. Further, GAN training can often lead to mode collapse [7], where the generator fails to generate samples that cover the entire data distribution and instead learns to produce a few unique samples. To overcome these challenges, a new class of generative model has been gaining attention among researchers based on Diffusion [126, 68]. These models are a type of generative probabilistic model that consists of two steps - in the first step, the forward diffusion process manipulates the data by steadily adding a small amount of random Gaussian noise over a series of time stamps until the data is destroyed. In the second step, a reverse diffusion process learns to train a model to restore the structure of the data by removing the noise over a series of time steps. The trained model then can sample information from a random Gaussian noise distribution and steadily denoise it over a series of time steps to generate the desired output. Due to their nature, diffusion models achieve high mode coverage than GAN, while the training of these networks is much simpler. At the same time, these models have shown extraordinary generating capabilities and beat GAN in tasks like image synthesis [41] and other guided generation tasks [101, 108, 110, 115]. In recent times diffusion models have gained popularity and show competitive results in image-to-image translation [115, 118], video generation problems [66, 69, 70], audio synthesis [31, 89, 59], and many others [157]. This makes the diffusion models an ideal choice for audio-driven video editing, which is mostly dominated by GAN-based approaches [30, 32, 140]. The main objectives of this study are -

- Explore the capabilities of diffusion models in speech-driven video editing tasks.
- Introduce the conditioning mechanism to the diffusion model based on the audio signal and other cues to maintain temporal stability.

6.1.3 Summary of Contribution

The work is presented in the article - Bigioi, Dan, Shubhajit Basak, Michał Stypułkowski, Maciej Zieba, Hugh Jordan, Rachel McDonnell, and Peter Corcoran. "Speech driven video editing via an audio-conditioned diffusion model." *Image and Vision Computing* (2024): 104911. A copy of the paper is attached at the end of this chapter.

The contributions of the authors for the research mentioned above work [19] as per the four major criteria discussed in section 1.4 is presented in the table 6.1. Though the primary work for these was carried out by Dan Bigioi, my contribution to these work are:

- Set up the evaluation experiments and perform the evaluation studies.
- Drafting the evaluation section in the manuscript.

Table 6.1 Author's Contribution to [19]

Contribution Criteria	Contribution Percentage
Ideation	DB 80%, PC 20%
Experiments & Implementations	DB 80%, SB 15%, HJ 5%
Manuscript Preparation	DB 80%, RM 5%, SB 5%, PC 10%
Background Work	DB 80%, PC 20%

To accomplish the above objectives, we proposed an unstructured end-to-end approach for speech-driven video editing using a denoising diffusion probabilistic model. Our work is based on the Palette [118] architecture, a denoising U-Net model originally trained for image-to-image translation tasks. We formulated our problem as an image inpainting task by masking out the bottom portion of the face. We particularly used a rectangular mask to hide the jaw contour, as we have found that if the jaw contour is visible to the network, it will learn to predict the lip movements based on the jaw alone, thus discarding the audio signal completely as noise. We conditioned the network on audio frames and trained to inpaint the lower half region of the face so that the lip and jaw movements are synchronized to the input audio signal. We train the network on both single and multi-speaker versions of the GRID dataset and demonstrate promising results despite access to a limited amount of data and training hardware. We compute the mel-spectrogram features from the conditioning audio and concatenate them with the image channel. As our approach works frame-by-frame to ensure the temporal consistency between the consecutive frames, we pass the preceding image frame and the previous, current, and future audio frame features while training. The experimental results show promising results and demonstrate that using a denoising diffusion model to do audio-driven video editing is feasible and produces reasonable results.

6.1.4 Discussion on Contribution

Through some initial experiments, we have shown that the denoising diffusion model can be applied successfully in audio-driven video editing tasks. Though it is able to perform reasonably to this task due to its nature, diffusion models are slow to train and infer. Our model is also no exception taking approximately 30 minutes per epoch to train the single-speaker model, 90 minutes per epoch for the multi-speaker one, and approximately 1 minute to generate one frame with 2000 diffusion steps on a single 32 GB V100 GPU. As future work, we are working on applying latent diffusion [115] that facilitates the training on latent space, thus shrinking the parameters of the model. We also need to incorporate new methodologies to infuse the audio features while training, as the current model performs badly in some specific syllables. Also, the multi-speaker model, model fails to keep the identity information while generating long videos. So work must be done to add additional constraints to add the identity information.

6.2 A Review of Benchmark Datasets and Training Loss Functions in Neural Depth Estimation

While we were working on the monocular depth estimation project, we surveyed the available real datasets that had depth information. As there is very limited monocular depth data available which has facial data, we studied each dataset and its attributes. We divided the datasets into five different categories - (i) people detection and action recognition, (ii) faces and facial pose, (iii) perception-based navigation (i.e., street signs, roads), (iv) object and scene recognition, and (v) medical applications. Also, we studied different data mixing strategies for neural depth estimation that can be found in the literature. Another key aspect of the monocular depth estimation task is the objective function. So we studied the common loss functions used in-depth estimation tasks and discussed their details, including their advantages and limitations.

The work is presented in the article - Khan, Faisal, Shahid Hussain, Shubhajit Basak, Mohamed Moustafa, and Peter Corcoran. "A Review of Benchmark Datasets and Training Loss Functions in Neural Depth Estimation." *IEEE Access* 9 (2021): 148479-148503. A copy of the paper is attached at the end of this chapter.

6.3 Copy of Published Works



Speech driven video editing via an audio-conditioned diffusion model

Dan Bigioi^{a,*}, Shubhajit Basak^a, Michał Stypułkowski^b, Maciej Zieba^{c,d}, Hugh Jordan^e, Rachel McDonnell^e, Peter Corcoran^a

^a University of Galway, Ireland

^b University of Wrocław, Poland

^c Wrocław University of Science and Technology, Poland

^d Tooploox, Poland

^e Trinity College Dublin, Ireland

ARTICLE INFO

Keywords:

Video editing
Talking head generation
Generative AI
Diffusion models
Dubbing

ABSTRACT

Taking inspiration from recent developments in visual generative tasks using diffusion models, we propose a method for end-to-end speech-driven video editing using a denoising diffusion model. Given a video of a talking person, and a separate auditory speech recording, the lip and jaw motions are re-synchronised without relying on intermediate structural representations such as facial landmarks or a 3D face model. We show this is possible by conditioning a denoising diffusion model on audio mel spectral features to generate synchronised facial motion. Proof of concept results are demonstrated on both single-speaker and multi-speaker video editing, providing a baseline model on the CREMA-D audiovisual data set. To the best of our knowledge, this is the first work to demonstrate and validate the feasibility of applying end-to-end denoising diffusion models to the task of audio-driven video editing. All code, datasets, and models used as part of this work are made publicly available here: <https://danbigioi.github.io/DiffusionVideoEditing/>.

1. Introduction

The idea behind audio-driven video editing is to provide a means to re-synchronise the lip and jaw movements of an actor in a video, in response to a new speech input signal. This new speech signal may come from the original speaker, or a voice actor. Regardless of the source of the input speech, a key objective is that the performance of the actor is never diminished. No matter how the lip and jaw movements change in response to the new audio, the facial expressions, and emotions portrayed by the actor should remain consistent with the original performance.

Achieving such seamless audio-driven video editing is an exciting prospect for the entertainment industry, one with the potential of being applied to movies, TV shows, live streaming, and even homemade content uploaded to platforms such as YouTube, TikTok, and others. Giving video content creators the ability and option to edit their work without having to go through time-consuming, and expensive re-shoots, allows them to work with a greater tolerance for error during filming.

Furthermore, the realisation of true audio-driven video editing would bring about a significant transformation in the world of cinema and television, allowing for more accessible and cost-effective dubbing of English-language movies/TV shows/videos into other languages and vice versa, allowing for the further democratisation of video content by making it more engaging and personalised for audiences worldwide. Recent advancements in deep learning and talking head generation techniques are bringing us closer to this exciting possibility, where audio and video will be seamlessly synchronised in real-time.

Current approaches for speech driven video editing, and the related task of talking head generation can be grouped into two distinct types: structured, and unstructured. Structured generation refers to techniques that use the speech signal to first extract an intermediate structural representation of the face (facial landmarks, 3D model expression parameters), before utilising it to render the photo-realistic frame [7,31,71,86,89]. On the other hand, unstructured generation approaches [18,29,73,88], utilise image reconstruction techniques to generate the photo-realistic frame directly in an end-to-end manner.

* Corresponding author.

E-mail addresses: d.bigioi@universityofgalway.ie (D. Bigioi), s.basak1@universityofgalway.ie (S. Basak), michal.stypulkowski@cs.uni.wroc.pl (M. Stypułkowski), maciej.zieba@pwr.edu.pl (M. Zieba), jordanhu@tcd.ie (H. Jordan), ramcdonn@tcd.ie (R. McDonnell), peter.corcoran@universityofgalway.ie (P. Corcoran).

<https://doi.org/10.1016/j.imavis.2024.104911>

Received 14 August 2023; Received in revised form 24 October 2023; Accepted 13 January 2024

Available online 14 January 2024

0262-8856/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Through the implementation of a divide-and-conquer strategy, structured approaches demonstrate the ability to generate videos of significantly higher quality compared to their unstructured counterparts. This strategy involves employing multiple models, each specialising in a distinct aspect of the generation process, ultimately contributing to a more accurate outcome during inference.

On the other hand, unstructured methods take a different approach, opting for a direct, end-to-end methodology that prioritises optimization of a single model in the pixel space. While this approach offers simplicity and streamlined processing, it inevitably involves a trade-off, sacrificing some of the accuracy achieved by structured methods. To address this, traditional methods frequently incorporate multiple loss functions, each designed to steer the optimization process towards refining the model's performance in the desired manner. Selecting the optimal combination of loss functions is a highly challenging task however, which involves striking a delicate balance between competing objectives to ensure the model captures essential features accurately, and converges during training.

Through the use of diffusion models, we introduce a method for training a stable end-to-end video editing model that overcomes the aforementioned challenges associated with unstructured methods. By leveraging an audio-conditioned diffusion model, our approach aims to enhance model stability during training, avoid common GAN pitfalls such as mode collapse, and, generate photorealistic high quality frames without relying on complicated multiple loss functions.

Diffusion models [16,24,48,62] are a relatively new class of generative model that have recently been gaining traction due to their strong performance on image synthesis tasks, often outperforming state-of-the-art GAN (Generative Adversarial Network) [20]-based methods [16]. Utilising conditioning signals such as text and even images, diffusion models have shown that they can be trained and conditioned towards generating a specific desired output at inference time with relative ease [55]. They achieve high mode coverage unlike GANs, maintain high sample quality, and are stable during training. These properties make them an ideal candidate for application towards the task of unstructured audio-driven video editing, a task that has thus far been dominated by GAN-based approaches [8,11,73].

We present an approach for automatic speech driven video editing using a denoising diffusion model. We utilise a U-Net backbone modifying it for the task of video editing, and introduce a feature concatenation mechanism for conditioning the network with information related to the previously generated frame in the sequence so that the network can generate temporally coherent frames. We further condition the network on speech by feeding spectrogram feature embeddings combined with the noise signal throughout the residual layers of the U-Net. To the best of our knowledge, this is the first work that applies denoising diffusion models to the task of audio-driven video editing. As part of this work, we state the following contributions to the field:

- A novel unstructured end-to-end approach for audio-driven video editing using a denoising diffusion model. We condition the network on speech and train it to modify the face such that the lip and jaw movements are synchronised to the conditioning audio signal on a frame-by-frame basis. We train both single, and multi-speaker proof-of-concept models using the GRID [14], and CREMA-D [6] datasets respectively, achieving strong proof-of-concept results when tested on unseen speakers. The project code, datasets, and trained models will be made freely available to the public.
- We demonstrate the applicability of our approach on the video editing task, achieving competitive results thanks to our conditional inpainting strategy which gathers information from previous frames and audio spectral embeddings, to generate the current frame. Our method achieves near state-of-the-art results when measured on traditional image quality metrics such as SSIM, PSNR, FID, and competitive SyncNet [13] lip synchronisation scores compared to other relevant methods from the field.

2. Related works

2.1. Audio driven video generation

Audio-driven video generation methods can generally be categorised by whether they are generated by leveraging an audio-driven structural representation of the face, or without.

There have been numerous approaches over the years relating to the former. Taylor et al. [70] and Karras et al. [32] among the first to apply machine learning techniques to the facial animation task, the former learning facial expression parameters of a 3D face model from phoneme labels, and the latter predicting 3D vertex positions of a face mesh from a speech audio window. Suwajanakorn et al. [68] trained a speaker specific network to output sparse mouth key-points, using them to modify videos of President Obama. Eskimez et al. [17] presented a recurrent architecture capable of taking in speech as input and outputting 2D landmark face co-ordinates, with Chen et al. [9] utilising cascaded GANs to translate those landmark features into photorealistic frames. Cudeiro et al. [15] introduced a 4D facial dataset, and trained a network to generate animations from speech with it. [5,40,74,89] generated intermediate landmark features from audio, also exploring the related task of extracting realistic headpose. Thies et al. [71] generated 3D facial expression parameters using features from a pretrained audio encoder, using these parameters to generate a photorealistic video via a neural renderer, with [64,76] following a similar approach but operating on videos instead. [7,82] presented methods to generate 3D face animation parameters, in addition to realistic head pose from speech, using these features to generate photorealistic frames. Ji et al. [31] approached the problem of video editing, generating emotion-controllable talking head portraits using both intermediate landmark structures, and 3D model parameters. [37,54,63,77,83,85] are other approaches from the literature which predict expression parameters from audio to drive a 3D face model.

What these approaches all have in common is that they use these intermediate structural representations as input to a separate neural rendering model which is typically trained as an image-to-image translation task to generate the final photo-realistic image frame. As of the date of this submission, GAN-based [20] approaches such as Pix2Pix [28], CycleGAN [91], and other variations have proved immensely popular for this task. However, diffusion-based techniques show big promise for the future, especially given recent developments in various image-to-image translation tasks [58].

Nonstructural/end-to-end methods on the other hand utilise latent feature learning and image reconstruction techniques to generate a photo-realistic video sequence from an input speech signal and reference image/video in an end-to-end manner. Approaches such as [8,18,29,36,44,50,66,73,87,88,90] have seen much success in recent times. Each of these approaches differs from the one used in this paper as they are all GAN/VAE (variational autoencoder) [34] based probabilistic methods while ours leverages a denoising diffusion model. While current end-to-end approaches suffer from low output resolution quality compared to structural methods, there is a lot of potential for improvement, especially by exploiting diffusion models' ability to synthesise high-quality samples while maintaining good mode coverage/diversity.

2.2. Diffusion models

Denoising diffusion models [62,65] have seen great success on a wide variety of different challenges, ranging from image-to-image translation tasks like inpainting, colourisation, image upscaling, uncropping [4,25,42,43,51,55,58,60], audio generation [10,27,33,35,38,49,69,79], text-based image generation [2,19,21,47,53,57,59], video generation [22,26,81,84], and many others. Recently, diffusion models have also been applied to the related task of talking head generation, with the works of [61,67], concurrent

approaches to our own. For a thorough review of diffusion models and all of their recent applications, we recommend [80].

Diffusion models are a class of generative probabilistic models that consist of two steps: 1) the forward diffusion process that destroys data by steadily adding small amounts of random Gaussian noise over a series of time steps until the data becomes a sample from a standard Gaussian distribution. 2) The reverse diffusion process where a denoising model is trained to restore structure in the data by steadily removing noise over a series of time steps. The trained model can then sample information from random Gaussian noise and steadily denoise it over a series of time steps to attain the desired output.

Sohl-Dickstein et al. [62] developed the first diffusion model and coined the term, followed by Ho et al. [24] combining denoising score matching with Langevin dynamics [65] and diffusion models to synthesise images. This ignited a steady interest in diffusion models, with Nichol et al. [48] showing that by making small adjustments to the diffusion process, they could sample data faster and achieve better log-likelihoods to models trained explicitly to minimise it with minimal impact to sample quality. They also found that training diffusion models with more computational power typically lead to better sample quality. Chen et al. [10] and Kong et al. [35] applied diffusion models to the task of audio synthesis, succeeding in generating high-quality samples. Dhariwal and Nichol [16] demonstrated that diffusion models beat GANs on image synthesis, also introducing the concept of “classifier guidance” for a conditional generation.

As diffusion models are trained under a single loss and do not rely on a discriminator, they are more stable during training and do not suffer from typical issues associated with training GANs such as mode collapse, and vanishing gradients. They produce high-quality output samples and display high mode coverage unlike GANs [78]. Despite these advantages, their sampling speed is slow due to the need to run the inverse diffusion process thousands of times on the same sample to denoise it

completely. Xiao et al. [78] and Rombach et al. [55] attempted at speeding up the sampling and training times associated with diffusion models with the former proposing a method to model the denoising distribution using a complex multi-modal distribution in order to facilitate larger diffusion steps, and the latter applying diffusion models in the latent space of a pre-trained autoencoder to reduce the complexity. This is an ongoing focus of research in the field, and it is a certainty that more works tackling the inference/training speed problem will emerge.

3. Materials and methods

A diffusion model is defined as having two steps, the forward diffusion process where the data is gradually destroyed, and the learned inverse diffusion process which reconstructs the data, and is used during training and inference. In our case, we condition a denoising U-Net on image and speech features to denoise a masked portion of the target frame into the desired output. A high-level overview of this process is depicted in Fig. 1.

3.1. Diffusion process

3.1.1. Forward diffusion process

As defined by [62], the forward diffusion process is a Markov chain that adds small amounts of noise to the data y over a predefined number of time steps T , until the data is completely destroyed at time step $t = T$. This state is represented as y_T with y_0 representing the data before any noise was added to it. The Markov chain is defined by:

$$q(y_{1:T}|y_0) := \prod_{t=1}^T q(y_t|y_{t-1}) \quad (1)$$

where at each step, Gaussian noise is added by:

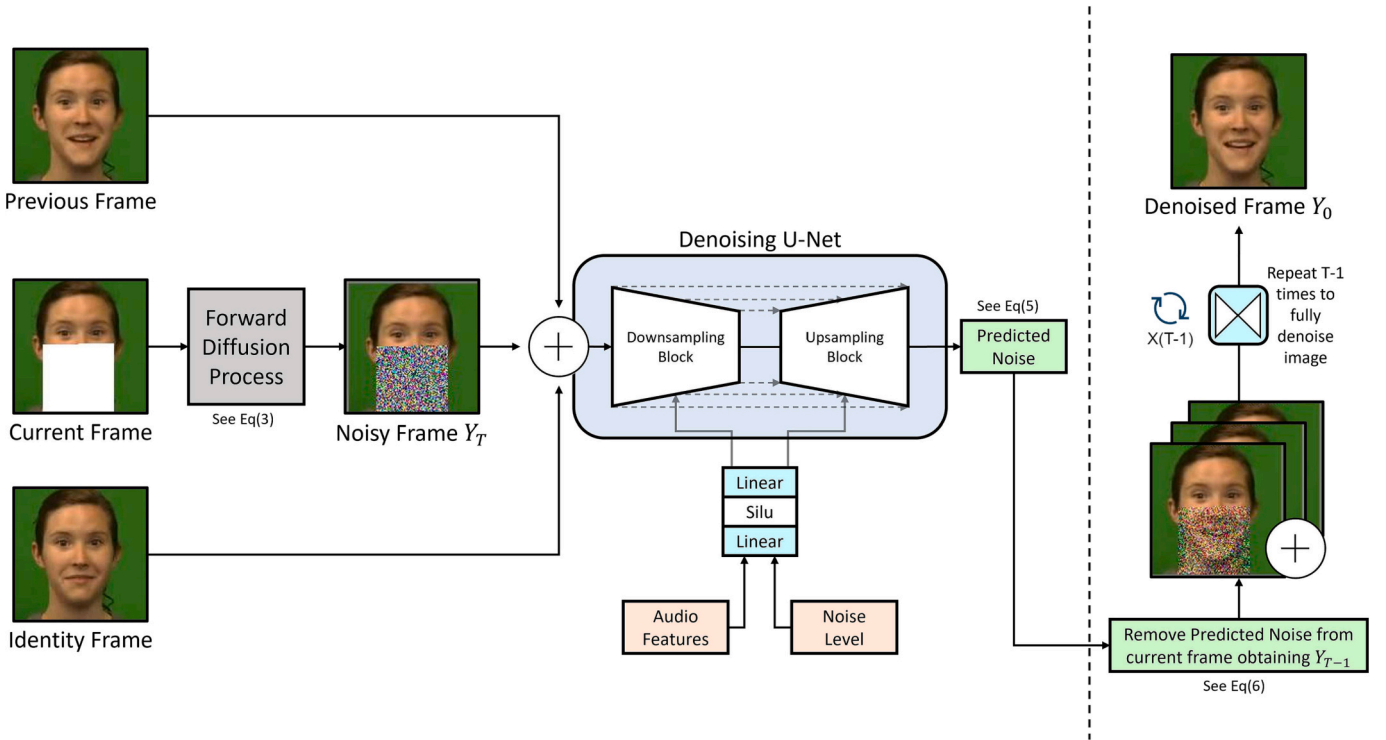


Fig. 1. High-level overview of the network architecture. Left of the dashed line indicates the training procedure, right of it depicts the inference procedure. \oplus represents the concatenation operator, and \rightarrow represents a skip connection. The current frame is passed through the forward diffusion process where the noise is computed and added to the masked region of the face, obtaining noisy frame Y_t (Eq. (3)). The previous and identity frames are then concatenated channel-wise to it, forming a $128 \times 128 \times 9$ feature and passed to the U-net directly. Audio features and noise level information are fed into the U-net through conditional residual blocks as described in Eq. (7), and depicted in Fig. 1. During inference, the predicted noise is removed from noisy image Y_t , obtaining Y_{t-1} . The previous and identity frames are concatenated to Y_{t-1} , and the process is repeated until the image is fully denoised (Eq. (6)).

$$q(y_t|y_{t-1}) := \mathcal{N}(y_t; \sqrt{\alpha_t}y_{t-1}, (1 - \alpha_t)I), \quad (2)$$

with $\alpha_t := (1 - \beta_t)$, representing the hyperparameters of our fixed noise scheduler. [24] show that it is possible to sample y_t at any step t in closed form:

$$q(y_t|y_0) := \mathcal{N}(y_t; \sqrt{\bar{\alpha}_t}y_0, (1 - \bar{\alpha}_t)I), \quad (3)$$

with $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$. This is an important observation, as it significantly speeds up the forward diffusion process, and can be used to train a model on the fly with random noise levels at each forward step.

3.1.2. Inverse diffusion process

Given a noisy image \bar{y} defined as:

$$\bar{y} := \sqrt{\bar{\alpha}_t}y_0 + \sqrt{1 - \bar{\alpha}_t}\varepsilon, \varepsilon \sim \mathcal{N}(0, I) \quad (4)$$

the goal of the Inverse diffusion process is to learn an algorithm that can denoise and restore the noisy image to its original image Y_0 . Following the approach in [58], we train a neural network $f_\theta(x, \bar{y}, \bar{\alpha}, \omega)$, a 2D U-Net [56], to predict the noise generated at time t , optimising the L_{simple} objective proposed by [24]:

$$\mathbb{E}_{t, y_0, \varepsilon} \left[\left\| f_\theta \left(x, \sqrt{\bar{\alpha}_t}y_0 + \sqrt{1 - \bar{\alpha}_t}\varepsilon, \bar{\alpha}, \omega \right) - \varepsilon \right\|^2 \right] \quad (5)$$

where x represents the identity and previous frame input to our network, \bar{y} the noisy image, $\bar{\alpha}$ the noise level, and ω the audio features. During training, we only calculate the loss for the masked region of the face to conserve computational resources, following the approach in [58].

Following [24], to run inference, each step of the inverse diffusion process can then be computed by:

$$y_{t-1} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} \left(y_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} f_\theta(x, y_t, \bar{\alpha}_t) \right) + \sqrt{1 - \alpha_t} \varepsilon_t, \quad (6)$$

where $\varepsilon \sim \mathcal{N}(0, I)$. The inverse diffusion step is then repeated T times. Please see Fig. 0 for a high-level view of our network architecture, and to better understand where each equation is used. For a more detailed discussion behind these equations, and how they are derived, please see [24,62,65].

3.2. Model architecture

Fig. 1 depicts the overall architecture of our model. We frame the problem of audio-driven video editing as a conditional inpainting task with a few key changes. Traditionally, inpainting is an image-to-image translation task where a neural network must learn to fill in a masked out region of the image with realistic content. For video editing, we must provide the network with additional context, to help guide its generation process. To do this, we split the conditioning step into two categories, frame-based, and audio-based conditioning.

3.2.1. Frame-based conditioning

For a given frame y^i extracted from a video consisting of frames (y^0, \dots, y^n) , our model takes three images as input: 1) the current *masked noisy frame* y_t^i that is to be inpainted, 2) the *previous frame* $y^{(i-1)}$ in the video sequence, and 3) a constant *identity frame* y^0 . As our approach is auto-regressive and works on a frame-by-frame basis, the purpose of the previous frame is to ensure that there is temporal stability between consecutive frames. Omitting it causes the model to output jittery, unstable frames. The identity frame is there to encourage the model not to deviate away from the target identity during the generation process, as so often is the case with auto-regressive models. While the identity frame can be omitted if training a single-speaker model with little to no adverse effects, we found that its inclusion was key to having a model that could generalise well to unseen subjects when training on multiple

identities. These three frames are concatenated channel-wise, and fed into the U-Net as an input feature of size [128x128x9], as depicted on the left hand side of Fig. 1.

3.2.2. Audio-based conditioning

For a given video sequence of frames (y^0, \dots, y^n) , there is a corresponding sequence of audio spectral features $(spec^0, \dots, spec^{2n})$ extracted from the original speech signal. Each audio feature spans a 40 ms window, overlapping every 20 ms. Details on how we compute these features are provided in Section 3.3. In order to provide the audio information to the network, we extract a window of audio from $(spec^{2i-2}$ to $spec^{2i+2})$ spanning 120 ms denoted as z^i that is centered around the current video frame y^i . We do this so that audio information from both the preceding and following frames is captured within the window to guarantee the accurate production of lip movements for plosive sounds (“p, t, k, b, d, g”) by taking into consideration that these lip movements precede the sound production. We then introduce this information to the U-net via the use of conditional residual blocks that condition the network on audio and noise level embeddings, scaling and shifting the hidden states of the U-net following the approach of [67]:

$$h_{s+1} = z_s^i (t_s GN(h_s) + t_b) + z_b^i \quad (7)$$

where h_s and h_{s+1} represent consecutive hidden states of the U-Net, $(z_s^i, z_b^i) = \text{MLP}(z^i)$, and $(t_s, t_b) = \text{MLP}(\bar{\alpha}_t)$. MLP represents a shallow neural network with a couple of linear layers separated by a SiLu() activation function, and GN is a group normalisation layer. This is shown detail in Fig. 2.

3.2.3. U-net set up

In order to denoise the current noisy frame, we use a denoising U-net [56], following the general architecture described by [58], which in turn is based on the model proposed by [24] with modifications inspired by the works of [16,60]. For this work we use a lightweight 128×128 version of the 256×256 U-net architecture described by [16], omitting the class conditioning mechanism. Like [58] we condition the model to generate the desired frames via the concatenation of the previous and identity frames to the masked frame. We drive the facial animation by sending spectral audio features throughout conditional residual blocks within the U-Net as detailed by [67], described by Eq. (7). We include all details related to our U-Net configuration in Table 1.

Table 1 displays the hyperparameters we use to train our diffusion model for the task of audio-driven video editing. We train two models, a single-speaker model trained on identity S1 of the GRID dataset, and a multispeaker model trained on the train set of the CREMA-D dataset. A notable difference between the two models is the use of attention. For the single-speaker model, we omitted it from the up/downsampling layers of the U-Net, using it only within the middle block in an effort to boost training speed. Despite this, we still obtain pleasing results, as shown both in Table 2, and in the videos provided as part of the supplementary materials. During our experiments, we discovered that the use of attention within the multi-speaker model was crucial for it to generalise well to both seen and unseen speakers. We apply it at resolutions of 32×32 within the up/downsampling layers of the U-Net. We provide more discussion on this in Section 4. To perform training we used a server of 4 32GB Nvidia V100 GPUs, allowing us a batch size of 40 per GPU. We trained the multi-speaker model for approximately 20 days, for a total of 735 epochs to achieve the results presented.

3.3. Data processing

3.3.1. Dataset

We rely on the GRID [14], and CREMA-D [6] audio-visual speech data sets to carry out the work in this paper. GRID is a multi-speaker data set consisting of 34 speakers (18 male, 16 female), with each speaker

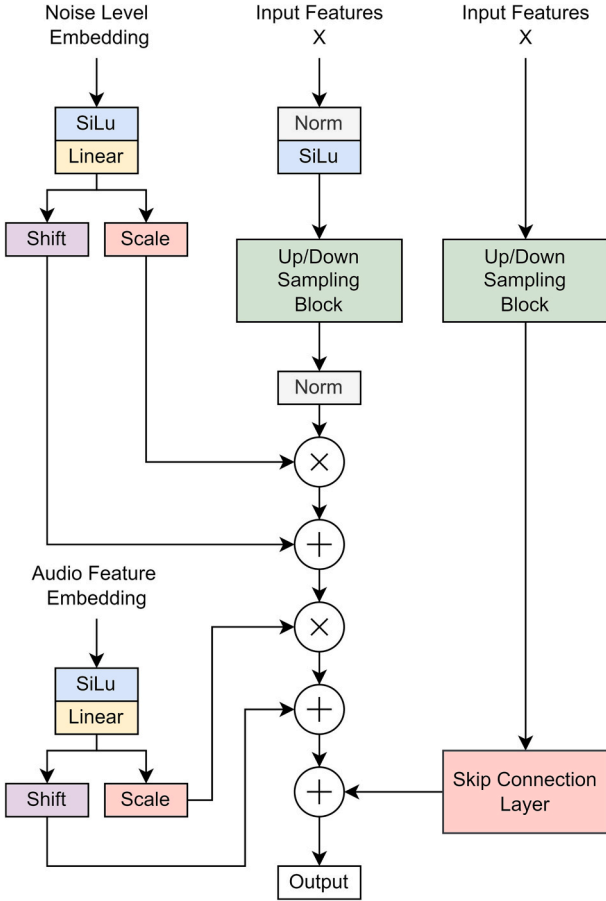


Fig. 2. High-level overview of the conditional residual blocks found within our U-Net architecture depicting the process in which we condition our model on noise level and audio embeddings. \oplus represents the addition operator, and \otimes represents the multiplication operator. Given the noise level embedding, we pass it through a SiLu activation function, and a linear layer, splitting the output into a scale and shift embedding. Meanwhile, the input features are normalised, and passed through a SiLu activation function before being passed through an up or downsampling block depending on where in the U-Net we are. The resultant features are then multiplied by the scale embeddings, and added to the shift embeddings that represent the noise level. As shown in the diagram, this process is repeated for the audio feature embeddings, with the final hidden state being added to the output of the skip connection layer. This sequence of events is described by Eq. (7).

Table 1
U-Net training hyperparameters.

	Single ID	Multi-ID
Image Size	128x128	128x128
Total Frames	73704	432000
Diffusion Steps	2000	1000
Noise Schedule	Linear	Cosine
Linear Start	$1e-06$	NA
Linear End	0.01	NA
Input Channels	10	10
Inner Channels	64	64
Channels Multiple	1, 2, 4, 8	1, 2, 3
Attention Resolution	NA	32
Res Blocks	2	2
Head Channels	32	32
Drop Out	0.2	0.2
Batch Size	10	40
Training Epochs	2000	735
Learning Rate	$5e-05$	$5e-05$

uttering 1000 short 6-word sentences. CREMA-D is a multi-speaker dataset consisting of 7442 talking head clips of 91 speakers from diverse ethnic backgrounds. As described by the [6], the racial/ethnic breakdown is as follows: 53 non-Hispanic Caucasian speakers, 8 Caucasian Hispanic speakers, 21 African American speakers, 1 African American Hispanic speaker, 7 Asian speakers, and 1 speaker of unspecified background.

We present two models: 1) A single speaker model trained on 950 videos from the speaker 1 of the GRID dataset, with the model's performance being evaluated on the remaining 50 videos on the task of video editing. 2) A multi-speaker model trained on a majority of the CREMA-D dataset totaling 432,000 individual samples (frames), with frames from videos of speaker identities 15, 20, 21, 30, 33, 52, 62, 81, 82, and 89 kept hidden from the model for testing and evaluation purposes.

3.3.2. Audio preprocessing

From each video within the GRID and CREMA-D datasets, we extract the audio files and resample them at 16Khz. From the audio we compute overlapping mel-spectrogram features with n-fft 2048, window length 640, hop length 320, and 256 mel bands. With these values, a 1-s audio feature has a shape [50,256] that can be easily aligned to a sequence of video frames.

3.3.3. Video preprocessing

First, we perform a 128×128 pixel crop centered around the face on every video frame. We do this by aligning the face in the video to the canonical face with a smoothing window of 7 frames, following the approach of [72]. We do this for two reasons: To get rid of any irrelevant background, and to reduce the image size to facilitate faster training and convergence speeds. In our initial experiments, we used an image size of 256×256 however the model was too expensive to train on our limited resources. It is worth noting that a video super-resolution technique such as [39] may be applied on top of our solution to achieve high-resolution samples.

Next, every video frame needs to have a rectangular region of the face masked out. Using an off-the-shelf facial landmark extractor [41], we extract facial landmark coordinates to determine the position of the jaw. Using this information, we mask out a rectangular portion of the face that covers a region just below the nose, as within Fig. 1. This face mask is computed and applied to the frames at train time within the data loader on the fly.

During training, it is critical to hide the speaker's jawline with a rectangular face mask. This is because the network can easily pick up on the strong correlation between lip and jaw movements, leading it to ignore the speech input entirely. By hiding the jawline, we compel the model to learn to generate lip movements based solely on the accompanying speech. As the diffusion process relies on a single loss function, applying the rectangular face-mask is the easiest way to prevent the frame-based input dominating over the speech input.

3.3.4. Audio video alignment

As described previously in Section 3.2, given a video sequence with frames (y^0, \dots, y^n) , there is a corresponding sequence of audio spectral features $(spec^0, \dots, spec^{2n})$ extracted from the original speech signal. Each audio feature spans a 40 ms window, overlapping every 20 ms. For any given frame Y^i , it is aligned to audio features spanned by $(spec^{2i-2}$ to $spec^{2i+2})$. To align the first and last video frames, we simply append silence to the start, and ends of their respective audio features. Care must be taken when choosing the audio window, too large and the network won't use the most meaningful information available to it, too small and there may not be enough context for the network to generate more complex lip movements caused by plosives.

Table 2

Quantitative comparison with previous works on image quality and lip synchronisation metrics. Most previous works we compare to require a driving video to guide the pose of the generated speaker. For these approaches (Actual) indicates whether we provided the ground truth video to their model in addition to the ground truth audio to generate the new video, while (Random) indicates that we used a random audio file instead. We report their results under both configurations to maintain fairness. For our models we also indicate how many diffusion timesteps were used to generate the frames during inference. We report results for 100, 500, and 1000 inference steps. † indicates that this metric was computed on the full frame. * indicates that these results are reported from their paper.

Method	LSE-C†	LSE-D↓	FID	SSIM†	PSNR†	CPBD
Ground Truth CREMA-D	5.45	8.12	–	–	–	–
EAMM (Actual)	3.98	8.92	22.52	0.74	29.43	0.1
EAMM (Random)	3.95	8.98	23.04	0.72	29.21	0.124
PC-AVS (Actual)	6.12	7.8	38.46	0.61	28.47	0.127
PC-AVS (Random)	6.07	7.82	40.05	0.59	28.42	0.11
SpeechDrivenAnimation	–	–	155.63	0.844*	27.98*	0.277*
Wav2Lip(Actual)	5.89	7.57	16.21	0.886	34.23	0.253
Wav2Lip(random)	5.6	7.89	20.23	0.872	34.04	0.247
Make It Talk	3.5	9.71	27.35	0.75	31.37	0.152
Ours (MultiSpeaker - 100)	3.53	9.74	2.362†	0.893	34.32	0.26
Ours (MultiSpeaker - 500)	3.5	9.68	2.13†	0.902	34.4	0.26
Ours (MultiSpeaker - 1000)	3.49	9.69	2.369†	0.863	34.12	0.242
Ours (Single Speaker)	4.98	7.59	2.312†	0.92	32.47	0.29

4. Results

In this section we present two models. A single-speaker video editing model trained on speaker S1 from the GRID dataset, and a multi-speaker model trained on the train-split of the CREMA-D dataset. We evaluate and compare our results to other recent audio-driven video generation methods, namely EAMM [30], PC-AVS [88], MakeItTalk [89], Speech Driven Animation [73], and Wav2Lip [50]. All models we test against are relevant end-to-end image-reconstruction based methods, except for MakeItTalk, a landmark-based method we compare against for reference purposes. We evaluate these models on the CREMA-D multispeaker test set, reporting their scores along with our own in Table 2. We generate the videos for each model using the official publicly available implementations with the recommended parameters.

As our models are trained explicitly for video editing, they generate only a small portion of the overall frame, while keeping the rest as is. Therefore, to maintain fairness, all metrics that rely on comparing the generated frame to the ground truth are computed only on the generated portion of the image. This limitation could also create bias in the perceptual metrics and readers should consider this when comparing our model scores to others within the literature.

We emphasise that the objective of this paper is to serve as a proof-of-concept demonstrating the potential of applying denoising diffusion models to the task of audio-driven video editing. As such, while we do not achieve state-of-the-art in some of the metrics we report, our results still show promising improvements over existing methods and highlight the potential of using denoising diffusion models for this task instead of traditional GAN-based methods.

4.1. Evaluation metrics

We use a number of objective metrics to measure the quality of our generated videos, allowing us to compare them directly to other state-of-the-art audio-driven video generation methods from the literature. We compute the following metrics:

- **SSIM [75]** (Structural Similarity Index Measure) †: SSIM evaluates the quality of an image by considering three key components: luminance, contrast, and structure. A higher SSIM value indicates a greater similarity between the images, implying that they are visually more alike.
- **PSNR** (Peak Signal to Noise Ratio) †: This measures the ratio between the maximum possible power of a signal and the power of the noise present in the signal. In the context of images, PSNR quantifies how much the quality of the image has degraded or been distorted compared to the original.

- **FID [23]** (Fréchet Inception Distance) ↓: This provides a measure of the similarity between the distribution of real images and the distribution of generated images. It captures both the quality and diversity of generated samples where a lower FID scores indicate better performance, suggesting that the generated images are close to the real data distribution.
- **CPBD [46]** (Cumulative Probability Blur Detection) †: This is a metric used to assess the overall blurriness of an image.
- **SyncNet [13,50] Confidence** (LSE-C) †: This the “average confidence score, where the higher the confidence, the better the audio-video correlation. A lower confidence score denotes that there are several portions of the video with completely out-of-sync lip movements”
- **SyncNet [13,50] Distance** (LSE-D) †: This is the average error measure “calculated in terms of the distance between the lip and audio representations, where a lower LSE-D indicates a higher audio-visual match, i.e., the speech and lip movements are in sync.”

We reiterate the point that in order to maintain fairness when computing the image quality metrics, we only compute them on the generated portion of the image where possible.

4.2. Single speaker

We train our single speaker model on identity S1 using data from the GRID audio-visual corpus [14]. There are 1000 videos in total, each of them roughly 3 s in length totaling about 50 min of audio-visual content for training. We train our model on 950 videos, withholding 50 of them for testing purposes. We train this model for 895 Epochs. As we mentioned previously, we did not use any attention layers within the up/downsampling blocks of this model, using it just within the middle block of the U-Net. We did this to save on training time, however, for stronger results we recommend using it, as we show within our multi-speaker model.

4.3. Multi-speaker

We train our multi-speaker model on all identities of the CREMA-D data set except for speakers 5, 20, 21, 30, 33, 52, 62, 81, 82, and 89, choosing to keep them hidden from the model for testing purposes. We train the model for 735 Epochs. There are a number of key changes we make to train the multi-speaker model. First, we use self-attention layers within the U-Net at the 32×32 resolution, as well as in the middle block. Second, we switch to a cosine noise schedule and decrease the number of diffusion steps taken by the model during training to 1000. Finally, we decrease the number of channel multiples to [1–3]. We also

experimented with training a model without attention in the up/downsampling blocks. It failed to converge on even train set identities. We speculate that increasing the number of inner channels used by our U-Net from 64 to 128 or 256 would significantly improve the results, as well as training the model for a longer amount of time. Please see Table 2 for a summary of our experiments and evaluations, compared to other popular works in the literature, and Section 4.4 for a detailed discussion surrounding the results.

4.4. Results discussion

Table 2 depicts the results our models score when tested on their unseen test sets versus other approaches in the literature. While the results we obtain are not state-of-the-art in all metrics, they successfully demonstrate that using a denoising diffusion model to do audio-driven video editing, is indeed quite feasible, and produces high-quality results comparable to other relevant methods in the literature.

The multi-speaker model generalises quite well to unseen speakers, scoring highly on image quality metrics, managing to outperform all other methods except for Wav2Lip on SSIM and CPBD. The single speaker model also achieving similarly strong results. We believe that this is due to the diffusion models inherent ability to model complex, high-dimensional data distributions, allowing it to learn the statistical properties of the dataset and generate images that are similar to those in the training set. Further, as diffusion models are trained to gradually remove noise from the target image over time, this may help it generate smoother, and more visually pleasing results than those generated by a GAN-based model which generates the frame in one shot. Within the context of audio-driven video editing, achieving visually pleasing results is a key requirement that our model fulfils. Please see the videos attached in the supplementary material for a visual comparison between our method and existing ones.

When evaluated on SyncNet [13] confidence (LSE-C) and distance (LSE-D) scores, our multi-speaker results are comparable to other popular methods from the literature, slightly outperforming MakeItTalk, but scoring lower than EAMM. PC-AVS and Wav2Lip score the highest in that order. Notably, their approaches significantly outperform the ground truth. We believe that this is because all other methods are specifically trained to optimise a loss function designed to penalise their models for poor lip synchronisation. In the case of PC-AVS and Wav2Lip, they both rely on a strong lip sync discriminator, to encourage their models to generate distinct, clear lip movements given speech. Our approach uses no such losses or discriminators, inherently learning the relationship between speech and lip movement during training. As such while our lip synchronisation scores on unseen speakers are lower, we offer a novel approach to the task as we do not explicitly train the model to improve lip synchronisation.

It is also worth noting that our single-speaker model performs very well on the synchronisation metrics mentioned above, leading us to speculate that with more time spent learning the data distribution, our

multi-speaker model could also theoretically achieve such results.

During inference, we noticed that the multi-speaker model occasionally struggled to maintain the identity consistent throughout the generation process, with the problem especially prevalent if there were extreme changes in head pose present in the original video. This is due to a buildup of small errors, as our approach is completely auto-regressive at inference time, relying entirely on just the previously generated frame, and identity frame to modify the current frame. Fig. 3 highlights one such instance of failure, and the phenomenon is noticeable in some of the videos we provide in our video abstract. We speculate that this could be alleviated in three ways 1) introducing small amounts of face warping on the previous frame during training in order to simulate the distortion that naturally occurs over the generation process. This would encourage the model to look at the identity frame in order to correct itself. 2) Simply train the model for longer. 3) Train on a more diverse dataset of speakers captured in unconstrained conditions such as Vox-Celeb or LRS.

When testing on identities seen by the network during training by replacing the original audio with a new one, the model achieves strong lip synchronisation, and the identity deviation seen when testing on unseen identities is significantly diminished, or simply does not occur over the course of the video. This problem is also non-existent in our single-speaker model.

We also observed that the multi-speaker model is highly sensitive to speaker volume, and intonation, especially when exposed to speech from unseen speakers. In instances where the speaker shouts or speaks loudly and clearly at the microphone, the lip movement is highly accurate and appears well-synchronised. When the volume is low, the speaker appears to be mumbling, and the full range of lip motion is not correctly generated. Analysing the synchronisation metrics confirmed this for us, with videos generated using audio labelled as being “angry” or “happy”, scoring significantly higher than instances where the portrayed emotion was “sad”, “fearful”, or “disappointed”. We suspect that this is due to our use of spectral feature embeddings when conditioning our network, and could be alleviated or significantly diminished with the use of a pretrained audio encoder for speech recognition. This is because such models are typically trained to extract the content from speech, disregarding information considered irrelevant such as pitch, or tone, and intonation.

5. Future work

5.1. Model speed and in the wild training

It is no secret that diffusion models are slow, both to train and to sample from. Our models are no exception, taking us approximately 6 min/epoch to train the single-speaker model, and 40 min/epoch for the multi-speaker one. We briefly experimented with training in the latent space to speed up training following the approach of [55], however, sample quality suffered, so we decided to operate in the pixel space. We

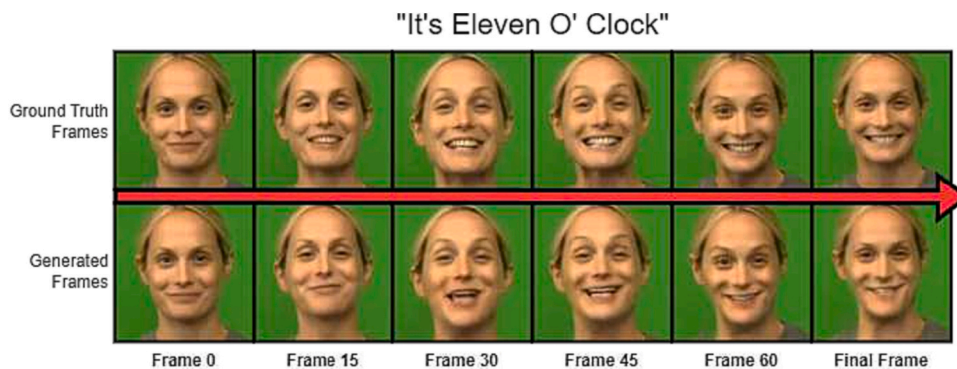


Fig. 3. Multi-speaker failure case: Over time the appearance of the speaker slowly drifts away from the original.

intend to revisit this however as improving our models training speed is a top priority for us as it would allow us to train on larger, more diverse, “in-the-wild” datasets such as VoxCeleb [45], or LRS [12].

Furthermore, our base model has plenty of scope for optimisation, and improvement. From utilising standard techniques such as mixed-precision training, improving our learning rate scheduling, and tweaking the number of layers and parameters in our model, we can improve the training duration of our approach, and facilitate learning on larger more diverse datasets to produce a more robust model.

5.2. Appearance consistency

As previously discussed, our multi-speaker model’s generated output appearance for unseen identities occasionally deviates from the original. To investigate this phenomenon, we intend to delve deeper into the underlying causes. Specifically, we will explore whether this effect is due to inadequate training or insufficient diversity in the training dataset, or a combination of both. By conducting a more detailed analysis, we hope to gain a better understanding of how to optimise our model’s performance for a wider range of identities. Further, we intend to fully train a model that utilises the face warping augmentation to determine whether this truly provides a positive impact on the generated samples.

5.3. Speech conditioning

We plan to explore the potential of conditioning our model with a wider range of speech features, such as experimenting with larger or smaller window sizes when computing spectral features or using pre-trained audio encoders such as Wav2Vec2 [3], Whisper [52], or DeepSpeech2 [1]. We believe that incorporating such features could potentially improve the lip synchronisation performance of our model and generate even more realistic and expressive lip movements.

6. Conclusion

Our results showcase the versatility of denoising diffusion models in capturing complex relationships between audio and video signals and generating coherent video sequences with accurate lip movements for the task of speech-driven video editing. We are encouraged by the strong performance achieved by our proof-of-concept approach, scoring highly on all tested metrics, comparable to existing state of the art in end-to-end video generation.

However, our work is not without limitations. The CREMA-D dataset is relatively small compared to other publicly available speech and video datasets, which limits the generalizability of our approach to other domains. Additionally, our approach requires a significant amount of computational resources and time to train. This is a challenge for real-time applications or for training on large-scale datasets.

We are confident that our work will inspire further research and development in this area, leading to more efficient and effective methods for speech-driven video editing. The practical applications that our work may enable in the future are exciting, ranging from on-demand real-time video editing applied to homemade content uploaded to websites such as YouTube or Tiktok, to big budget Hollywood movie productions, allowing them to save time and money on costly re-shoots. Major streaming services such as Netflix also stand to benefit immensely from effective video-editing technology as it may provide them the ability to dub content quickly, and effectively, expanding the global reach of their services to audiences across the world. With the continuing advancements in machine learning and computer vision, we believe that denoising diffusion models will play an increasingly important role in enabling high-quality and immersive multimedia experiences that can better reflect the diversity and richness of human communication.

CRedit authorship contribution statement

Dan Bigioi: Conceptualization, Investigation, Data curation, Methodology, Software, Writing – original draft. **Shubhajit Basak:** Data curation, Investigation, Validation, Writing – review & editing. **Michał Stypulkowski:** Data curation, Methodology, Writing – review & editing, Supervision. **Maciej Zieba:** Writing – review & editing, Supervision. **Hugh Jordan:** Validation, Resources. **Rachel McDonnell:** Conceptualization, Writing – review & editing, Supervision. **Peter Corcoran:** Conceptualization, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Dan Bigioi reports financial support was provided by Science Foundation Ireland. Hugh Jordan reports financial support was provided by Science Foundation Ireland. Peter Corcoran reports a relationship with Xperi Corporation that includes: consulting or advisory. Tooploox reports a relationship with Maciej Zieba that includes: consulting or advisory.

Data availability

I have shared all code and data related to the paper at the following link: <https://github.com/DanBigioi/DiffusionVideoEditing>

Acknowledgements

This work has the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224, and the ADAPT Centre (Grant 13/RC/2106).

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.imavis.2024.104911>.

References

- [1] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al., Deep speech 2: end-to-end speech recognition in english and mandarin, in: International Conference on Machine Learning, PMLR, 2016, pp. 173–182.
- [2] O. Avrahami, D. Lischinski, O. Fried, Blended diffusion for text-driven editing of natural images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 18208–18218.
- [3] A. Baevski, Y. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: a framework for self-supervised learning of speech representations, Adv. Neural Inf. Proces. Syst. 33 (2020) 12449–12460.
- [4] G. Batzolis, J. Stanczuk, C.B. Schönlieb, C. Etmann, Conditional image generation with score-based diffusion models, arXiv (2021) preprint arXiv:2111.13606.
- [5] S. Biswas, S. Sinha, D. Das, B. Bhowmick, Realistic talking face animation with speech-induced head motion, in: Proceedings of the Twelfth Indian Conference on Computer Vision, Graphics and Image Processing, 2021, pp. 1–9.
- [6] H. Cao, D.G. Cooper, M.K. Keutmann, R.C. Gur, A. Nenkova, R. Verma, Crema-d: crowd-sourced emotional multimodal actors dataset, IEEE Trans. Affect. Comput. 5 (2014) 377–390, <https://doi.org/10.1109/TAFFC.2014.2336244>.
- [7] L. Chen, G. Cui, C. Liu, Z. Li, Z. Kou, Y. Xu, C. Xu, Talking-head generation with rhythmic head motion, in: European Conference on Computer Vision, Springer, 2020, pp. 35–51.
- [8] L. Chen, Z. Li, R.K. Maddox, Z. Duan, C. Xu, Lip movements generation at a glance, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 520–535.
- [9] L. Chen, R.K. Maddox, Z. Duan, C. Xu, Hierarchical cross-modal talking face generation with dynamic pixel-wise loss, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7832–7841.
- [10] N. Chen, Y. Zhang, H. Zen, R.J. Weiss, M. Norouzi, W. Chan, Wavegrad: estimating gradients for waveform generation, arXiv (2020) preprint arXiv:2009.00713.
- [11] S. Chen, Z. Liu, J. Liu, L. Wang, Talking head generation driven by speech-related facial action units and audio-based on multimodal representation fusion, arXiv (2022) preprint arXiv:2204.12756.

- [12] J.S. Chung, A. Senior, O. Vinyals, A. Zisserman, Lip reading sentences in the wild, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 3444–3453.
- [13] J.S. Chung, A. Zisserman, Out of time: automated lip sync in the wild, in: Workshop on Multi-View Lip-Reading, ACCV, 2016.
- [14] M. Cooke, J. Barker, S. Cunningham, X. Shao, An audio-visual corpus for speech perception and automatic speech recognition, *J. Acoust. Soc. Am.* 120 (2006) 2421–2424.
- [15] D. Cudeiro, T. Bolkart, C. Laidlaw, A. Ranjan, M.J. Black, Capture, learning, and synthesis of 3d speaking styles, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10101–10111.
- [16] P. Dhariwal, A. Nichol, Diffusion models beat gans on image synthesis, in: Advances in Neural Information Processing Systems 34, 2021, pp. 8780–8794.
- [17] S.E. Eskimez, R.K. Maddox, C. Xu, Z. Duan, Generating talking face landmarks from speech, in: International Conference on Latent Variable Analysis and Signal Separation, Springer, 2018, pp. 372–381.
- [18] S.E. Eskimez, R.K. Maddox, C. Xu, Z. Duan, End-to-end generation of talking faces from noisy speech, in: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 1948–1952.
- [19] W.C. Fan, Y.C. Chen, D. Chen, Y. Cheng, L. Yuan, Y.C.F. Wang, Frido: feature pyramid diffusion for complex scene image synthesis. Proceedings of the AAAI Conference on Artificial Intelligence 37, 2023.
- [20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, *Commun. ACM* 63 (2020) 139–144.
- [21] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, B. Guo, Vector quantized diffusion model for text-to-image synthesis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10696–10706.
- [22] W. Harvey, S. Naderiparizi, V. Masrani, C. Weillbach, F. Wood, Flexible diffusion modeling of long videos, *Advances in Neural Information Processing Systems* 35 (2022) 27953–27965.
- [23] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, in: Advances in Neural Information Processing Systems, 2017, p. 30.
- [24] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, *Adv. Neural Inf. Process. Syst.* 33 (2020) 6840–6851.
- [25] J. Ho, C. Saharia, W. Chan, D.J. Fleet, M. Norouzi, T. Salimans, Cascaded diffusion models for high fidelity image generation, *J. Mach. Learn. Res.* 23 (2022) 1–47.
- [26] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, D.J. Fleet, Video diffusion models, arXiv (2022) preprint arXiv:2204.03458.
- [27] R. Huang, Z. Zhao, H. Liu, J. Liu, C. Cui, Y. Ren, Prodiff: progressive fast diffusion model for high-quality text-to-speech, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 2595–2605.
- [28] P. Isola, J.Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (2017) 1125–1134.
- [29] A. Jamaludin, J.S. Chung, A. Zisserman, You said that?: Synthesizing talking faces from audio, *Int. J. Comput. Vis.* 127 (2019) 1767–1779.
- [30] X. Ji, H. Zhou, K. Wang, Q. Wu, W. Wu, F. Xu, X. Cao, Eamm: one-shot emotional talking face via audio-based emotion-aware motion model, in: ACM SIGGRAPH 2022 Conference Proceedings, 2022, pp. 1–10.
- [31] X. Ji, H. Zhou, K. Wang, W. Wu, C.C. Loy, X. Cao, F. Xu, Audio-driven emotional video portraits, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14080–14089.
- [32] T. Karras, T. Aila, S. Laine, A. Herva, J. Lehtinen, Audio-driven facial animation by joint end-to-end learning of pose and emotion, *ACM Trans. Graph. (TOG)* 36 (2017) 1–12.
- [33] S. Kim, H. Kim, S. Yoon, Guided-tts 2: a diffusion model for high-quality adaptive text-to-speech with untranscribed data, arXiv (2022) preprint arXiv:2205.15370.
- [34] D.P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv (2013) preprint arXiv:1312.6114.
- [35] Z. Kong, W. Ping, J. Huang, K. Zhao, B. Catanzaro, Diffwave: a versatile diffusion model for audio synthesis, arXiv (2020) preprint arXiv:2009.09761.
- [36] N. Kumar, S. Goel, A. Narang, M. Hasan, Robust one shot audio to video generation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 770–771.
- [37] A. Lahiri, V. Kwatra, C. Frueh, J. Lewis, C. Bregler, Lipsync3d: data-efficient learning of personalized 3d talking faces from video using pose and lighting normalization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2755–2764.
- [38] A. Levkovitch, E. Nachmani, L. Wolf, Zero-shot voice conditioning for denoising diffusion tts models, arXiv (2022) preprint arXiv:2206.02246.
- [39] C. Liu, H. Yang, J. Fu, X. Qian, Learning trajectory-aware transformer for video super-resolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5687–5696.
- [40] Y. Lu, J. Chai, X. Cao, Live speech portraits: real-time photorealistic talking-head animation, *ACM Trans. Graph. (TOG)* 40 (2021) 1–17.
- [41] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C. L. Chang, M.G. Yong, J. Lee, et al., Mediapipe: a framework for building perception pipelines, arXiv (2019) preprint arXiv:1906.08172.
- [42] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, L. Van Gool, Repaint: inpainting using denoising diffusion probabilistic models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11461–11471.
- [43] C. Meng, Y. Song, J. Song, J. Wu, J.Y. Zhu, S. Ermon, Sdedit: image synthesis and editing with stochastic differential equations, arXiv (2021) preprint arXiv: 2108.01073.
- [44] G. Mittal, B. Wang, Animating face using disentangled audio representations, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 3290–3298.
- [45] A. Nagrani, J.S. Chung, A. Zisserman, Voxceleb: a large-scale speaker identification dataset, arXiv (2017) preprint arXiv:1706.08612.
- [46] N.D. Narvekar, L.J. Karam, A no-reference image blur metric based on the cumulative probability of blur detection (CPBD), *IEEE Trans. Image Process.* 20 (2011) 2678–2683.
- [47] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, M. Chen, Glide: Towards photorealistic image generation and editing with text-guided diffusion models. International Conference on Machine Learning, PMLR, 2022, pp. 16784–16804.
- [48] A.Q. Nichol, P. Dhariwal, Improved denoising diffusion probabilistic models, in: International Conference on Machine Learning, PMLR, 2021, pp. 8162–8171.
- [49] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, M. Kudinov, Grad-tts: a diffusion probabilistic model for text-to-speech, in: International Conference on Machine Learning, PMLR, 2021, pp. 8599–8608.
- [50] K. Prajwal, R. Mukhopadhyay, V.P. Namboodiri, C. Jawahar, A lip sync expert is all you need for speech to lip generation in the wild, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 484–492.
- [51] K. Preechakul, N. Chatthee, S. Wizadwongsa, S. Suwajanakorn, Diffusion autoencoders: toward a meaningful and decodable representation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10619–10629.
- [52] A. Radford, J.W. Kim, T. Xu, G. Brockman, C. McLeavey, I. Sutskever, Robust speech recognition via large-scale weak supervision, in: International Conference on Machine Learning, PMLR, 2023, July, pp. 28492–28518.
- [53] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, M. Chen, Hierarchical text-conditional image generation with clip latents, arXiv (2022) preprint arXiv:2204.06125.
- [54] A. Richard, M. Zollhöfer, Y. Wen, F. De la Torre, Y. Sheikh, Meshtalk: 3d face animation from speech using cross-modality disentanglement, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 1173–1182.
- [55] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.
- [56] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and computer-Assisted Intervention, Springer, 2015, pp. 234–241.
- [57] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, K. Abernath, Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 22500–22510.
- [58] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, M. Norouzi, Palette: image-to-image diffusion models, in: ACM SIGGRAPH 2022 Conference Proceedings, 2022, pp. 1–10.
- [59] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S.K.S. Ghasemipour, B. K. Ayan, S.S. Mahdavi, R.G. Lopes, T. Salimans, Photorealistic text-to-image diffusion models with deep language understanding, *Advances in Neural Information Processing Systems* 35 (2022) 36479–36494.
- [60] C. Saharia, J. Ho, W. Chan, T. Salimans, D.J. Fleet, M. Norouzi, Image super-resolution via iterative refinement, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.
- [61] S. Shen, W. Zhao, Z. Meng, W. Li, Z. Zhu, J. Zhou, J. Lu, DiffTalk: crafting diffusion models for generalized audio-driven portraits animation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 1982–1991.
- [62] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: International Conference on Machine Learning, PMLR, 2015, pp. 2256–2265.
- [63] L. Song, B. Liu, G. Yin, X. Dong, Y. Zhang, J.X. Bai, Tacr-net: editing on deep video and voice portraits, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 478–486.
- [64] L. Song, W. Wu, C. Qian, R. He, C.C. Loy, Everybody’s talkin’: Let me talk as you want, *IEEE Trans. Inf. Forens. Secur.* 17 (2022) 585–598.
- [65] Y. Song, S. Ermon, Generative modeling by estimating gradients of the data distribution, in: Advances in Neural Information Processing Systems 32, 2019.
- [66] Y. Song, J. Zhu, D. Li, A. Wang, H. Qi, Talking face generation by conditional recurrent adversarial network, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 919–925.
- [67] M. Stypulkowski, K. Vougioukas, S. He, M. Zięba, S. Petridis, M. Pantic, Diffused heads: Diffusion models beat gans on talking-face generation, Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (2024) 5091–5100.
- [68] S. Suwajanakorn, S.M. Seitz, I. Kemelmacher-Shlizerman, Synthesizing obama: learning lip sync from audio, *ACM Trans. Graph. (TOG)* 36 (2017) 1–13.
- [69] J. Tae, H. Kim, T. Kim, Edits: score-based editing for controllable text-to-speech, arXiv (2021) preprint arXiv:2110.02584.
- [70] S. Taylor, T. Kim, Y. Yue, M. Mahler, J. Krahe, A.G. Rodriguez, J. Hodgins, I. Matthews, A deep learning approach for generalized speech animation, *ACM Trans. Graph. (TOG)* 36 (2017) 1–11.
- [71] J. Thies, M. Elgharib, A. Tewari, C. Theobalt, M. Nießner, Neural voice puppetry: audio-driven facial reenactment, in: European Conference on Computer Vision, Springer, 2020, pp. 716–731.

- [72] K. Vougioukas, S. Petridis, M. Pantic, End-to-end speech-driven facial animation with temporal gans, *ArXiv* (2018) (abs/1805.09313).
- [73] K. Vougioukas, S. Petridis, M. Pantic, Realistic speech-driven facial animation with gans, *Int. J. Comput. Vis.* 128 (2020) 1398–1413.
- [74] S. Wang, L. Li, Y. Ding, C. Fan, X. Yu, Audio2head: audio-driven one-shot talking-head generation with natural head motion, *arXiv* (2021) preprint arXiv:2107.09293.
- [75] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (2004) 600–612.
- [76] X. Wen, M. Wang, C. Richardt, Z.Y. Chen, S.M. Hu, Photorealistic audio-driven video portraits, *IEEE Trans. Vis. Comput. Graph.* 26 (2020) 3457–3466.
- [77] H. Wu, J. Jia, H. Wang, Y. Dou, C. Duan, Q. Deng, Imitating arbitrary talking style for realistic audio-driven talking face synthesis, in: *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 1478–1486.
- [78] Z. Xiao, K. Kreis, A. Vahdat, Tackling the generative learning trilemma with denoising diffusion gans, *arXiv* (2021) preprint arXiv:2112.07804.
- [79] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. "Diffsound: Discrete diffusion model for text-to-sound generation, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2023).
- [80] Ling Yang, Zhilong Zhang, Yang Song, Hong Shenda, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, Yang Ming-Hsuan, Diffusion models: A comprehensive survey of methods and applications, *ACM Computing Surveys* 56 (4) (2023) 1–39.
- [81] R. Yang, P. Srivastava, S. Mandt, Diffusion probabilistic modeling for video generation, *arXiv* (2022) preprint arXiv:2203.09481.
- [82] R. Yi, Z. Ye, J. Zhang, H. Bao, Y.J. Liu, Audio-driven talking face video generation with learning-based personalized head pose, *arXiv* (2020) preprint arXiv:2002.10137.
- [83] C. Zhang, Y. Zhao, Y. Huang, M. Zeng, S. Ni, M. Budagavi, X. Guo, Facial: synthesizing dynamic talking face with implicit attribute learning, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3867–3876.
- [84] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, Z. Liu, Motiondiffuse: text-driven human motion generation with diffusion model, *arXiv* (2022) preprint arXiv:2208.15001.
- [85] Z. Zhang, L. Li, Y. Ding, C. Fan, Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3661–3670.
- [86] R. Zhao, T. Wu, G. Guo, Sparse to dense motion transfer for face image animation, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1991–2000.
- [87] H. Zhou, Y. Liu, Z. Liu, P. Luo, X. Wang, Talking face generation by adversarially disentangled audio-visual representation, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 9299–9306.
- [88] H. Zhou, Y. Sun, W. Wu, C.C. Loy, X. Wang, Z. Liu, Pose-controllable talking face generation by implicitly modularized audio-visual representation, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4176–4186.
- [89] Y. Zhou, X. Han, E. Shechtman, J. Echevarria, E. Kalogerakis, D. Li, Makeltalk: speaker-aware talking-head animation, *ACM Trans. Graph. (TOG)* 39 (2020) 1–15.
- [90] H. Zhu, H. Huang, Y. Li, A. Zheng, R. He, Arbitrary talking face generation via attentional audio-visual coherence learning, in: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 2362–2368.
- [91] J.Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.

Received October 16, 2021, accepted October 30, 2021, date of publication November 2, 2021, date of current version November 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3124978

A Review of Benchmark Datasets and Training Loss Functions in Neural Depth Estimation

FAISAL KHAN¹, SHAHID HUSSAIN², SHUBHAJIT BASAK³, (Graduate Student Member, IEEE),
MOHAMED MOUSTAFA¹, (Member, IEEE), AND PETER CORCORAN¹, (Fellow, IEEE)

¹Department of Electronic Engineering, College of Science and Engineering, National University of Ireland Galway, Galway, H91 TK33 Ireland

²Data Science Institute, National University of Ireland Galway, Galway, H91 TK33 Ireland

³School of Computer Science, National University of Ireland Galway, Galway, H91 TK33 Ireland

Corresponding author: Faisal Khan (f.khan4@nuigalway.ie)

This work was supported in part by the College of Science and Engineering, National University of Ireland Galway, Galway, Ireland; in part by the Xperi Galway Block 5 Parkmore East Business Park, Galway, Ireland; and in part by the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant 18/CRT/6224.

ABSTRACT In many applications, such as robotic perception, scene understanding, augmented reality, 3D reconstruction, and medical image analysis, depth from images is a fundamentally ill-posed problem. The success of depth estimation models relies on assembling a suitably large and diverse training dataset and on the selection of appropriate loss functions. It is critical for researchers in this field to be made aware of the wide range of publicly available depth datasets along with the properties of various loss functions that have been applied to depth estimation. Selection of the right training data combined with appropriate loss functions will accelerate new research and enable better comparison with state-of-the-art. Accordingly, this work offers a comprehensive review of available depth datasets as well as the loss functions that are applied in this problem domain. These depth datasets are categorised into five primary categories based on their application, namely (i) people detection and action recognition, (ii) faces and facial pose, (iii) perception-based navigation (i.e., street signs, roads), (iv) object and scene recognition, and (v) medical applications. The important characteristics and properties of each depth dataset are described and compared. A mixing strategy for depth datasets is presented in order to generalise model results across different environments and use cases. Furthermore, depth estimation loss functions that can help with training deep learning depth estimation models across different datasets are discussed. State-of-the-art deep learning-based depth estimation methods evaluations are presented for three of the most popular datasets. Finally, a discussion about challenges and future research along with recommendations for building comprehensive depth datasets will be presented as to help researchers in the selection of appropriate datasets and loss functions for evaluating their results and algorithms.

INDEX TERMS Datasets, depth datasets, depth loss function, deep learning, depth estimation.

I. INTRODUCTION

Depth estimation, the process of preserving 3D information of a scene using 2D information acquired by camera, can prove beneficial for many challenging computer-vision applications. Examples include human-machine interaction, robotics, augmented reality, object detection, pose estimation, semantic segmentation, and 3D reconstruction. Having access to ground truth depth information is valuable for developing robust guidance systems in autonomous vehicles, environment reconstruction, security, and image understanding

The associate editor coordinating the review of this manuscript and approving it for publication was Vicente Alarcon-Aquino¹.

where it is desirable to determine the primary objects and region with the imaged scene.

To this end, various methods have been developed to capture depth measurements as well as to research depth estimation using monocular or multi-view solutions, which aim to find the distance between scene objects and camera from a single or multiple point(s) of view relying on one or more images.

This study presents a detailed overview of depth datasets, depth loss functions, and their applications in the field of computer vision. Starting with a brief description (literature, definitions), datasets are analyzed in terms of citations, and then depth datasets are classified according to their

applications, the important characteristics and properties of each depth dataset are described and compared. Afterwards, depth-based loss functions and a mixing strategy for depth datasets are briefly discussed. Finally, state-of-the-art deep learning-based depth estimation methods evaluations and discussion about challenges and future research along with recommendations for building comprehensive depth datasets are presented.

A. APPLICATION CLASSES OF DEPTH DATASET

Datasets play a crucial role in scientific research, specifically for artificial intelligence models, datasets are the building block for analysing the performance and validating their results. Different datasets contain data captured in different environments (e.g., indoor vs outdoor scenes), of different objects, depth annotation types (relative, absolute, dense, sparse), accuracies (laser stereo, time-of-flight, synthetic data, structure-from-motion, human annotation), image quality, size, and camera settings. Every dataset has its own features and related problems and biases [1]. Large dataset collections from internet sources have many issues including quality of images, accuracy, and unknown camera parameters [2], [3]. High quality datasets can play an important role at enabling researchers to develop depth solutions for specific computer vision depth problems [4], [5].

Depth datasets are classified into various categories depending on particular task-based applications (i.e., indoor/outdoor, portrait/driver, half/full body scene, indoor small room, large street scene, large indoor scene, landscape/cityscape, and medical). A map of per-pixel data containing depth-related information is referred to as depth data. A depth data object incorporates a disparity or depth map and offers conversion methods, focus information, and camera calibration data to help with rendering and computer vision applications.

Structured light cameras, which give dense depth maps up to 10 meters, are commonly used to collect indoor depth information. They work by projecting a sequence of known patterns onto an object, and the deformation resulting from the object's shape is then observed through a camera from some other direction. Depth information can then be extracted from the observed distortion's disparity from the original projected pattern. The original Kinect sensor, also called Kinect v1, along with the Asus Xtion Pro, utilize this approach for depth capture [6]. Another commonly used technique is time-of-flight cameras, such as the Kinect v2, which relies on measuring the round-trip time for an emitted light using a sensor array and illumination unit [6]. Indoor places include locations such as offices, labs, corridors, study rooms, laboratories, and kitchens. Visual localization allows for intriguing applications like robot navigation and augmented reality by estimating the precise location of a camera. This is particularly useful in indoor environments where other localization technologies, such as Global Navigation Satellite System (GNSS), fail. Indoor spaces impose interesting tasks on visual localization methods (i.e., texture-less surfaces,

occlusions due to people, large view-point changes, repetitive textures, and low light).

Outdoor depth datasets are typically collected with a specific application in mind such as autonomous vehicles and generally captured with customized sensor arrays consisting of multi or monocular cameras and Light Detection and Ranging (LiDAR) scanners. Outdoor place categories include street signs, forests, indoor/outdoor parking lots, urban areas, roads, residential areas, and coast areas. The primary applications of outdoor depth datasets involve perception tasks in the context of autonomous vehicles, semantic scene understanding, and 3D reconstruction.

Human faces are one of the most prevalent features in images, and thus are a key part of a lot of computer vision tasks. It is widely known in human skeletal anatomy that the eye-separation in a human face fall within a small range, thus given information of a camera's field-of-view, it is feasible to calculate the distance-to-camera of a human subject with reasonable accuracy [7]. Human facial depth datasets include facial images, depth maps, images of the visible light spectrum (i.e., RGB), 3D depth maps, and head pose information. Deep neural networks can be trained to detect age, face, and gender using facial depth datasets, or to pick the optimum type of image for a specific task, such as facial recognition. It is also feasible to utilize data from people in random and frontal orientations to see if a facial recognition system can recognize faces from different perspectives [7], [8]. The face recognition system is typically divided into two different tasks in the computer vision field such face identification and face verification. The former is based on a one-to-many comparison to recognize the best match between a given face and a set of possibilities. While the latter uses a one-to-one comparison and can find whether the input item is of the same person's face or not.

Depth datasets created for a medical application consist of multi-view frames, video, RGB, depth maps, calibration parameters, 2D and/or 3D pose annotations, and human bounding boxes. The data generated during surgeries can be used for medical image analysis and machine learning to observe, analyze, model and support staff activities and clinician in the operating rooms.

Ideally researchers should combine multiple datasets during training, validation, and testing to improve generalization, but care is needed when combining datasets with differing characteristics. The design and building blocks of the network are important, but the performance of the network is mostly determined by how it is trained which requires a diverse dataset and a suitable loss function.

B. LOSS FUNCTIONS FOR DEPTH DATASETS

Another way to improve the deep network's training results is by introducing an appropriate loss function. The loss function calculates the network output's variance from the estimated output which is used to adjust the parameters of the deep network. This is achieved by backpropagating the error calculated using the loss function to the first layer in the training

process, changing the network's weights at each iteration. In the literature, several losses, architectures, and experimental conditions are given, but it is difficult to determine their relative influence on performance. An in-depth study is proposed of different losses and experimental situation for depth regression in this research.

A deep network must have a loss function. The loss function must be differentiable because of the back-propagation stage used in deep learning systems, which relies on propagating the gradients of the model's error from the output layer back towards the first layer. An in-depth study of various loss functions for depth regression is proposed that can be used for both short and long-range depth datasets.

C. RESEARCH CONTRIBUTIONS

This review aims to collect the available depth image datasets using bibliometric research by providing detailed information on the available datasets. Additionally, an easy and brief description is presented for each of the datasets to provide a basis for predicting depth estimation trends and explores their sub-areas; dataset popularity helps in identifying study areas that receive less attention.

The main scope of this study is to make it easier to navigate among the depth datasets and common loss functions that are frequently used in the depth estimation research. A list of popular datasets is compiled by looking through the publications indexed by the web of science library and IEEE Explore, as well as doing searches utilizing online search engines. These datasets are classified into different use case categories and present their detailed description such as (camera tracking, scene reconstruction, tracking, semantic, pose, video and recognition, streets, people i.e., identity recognition/faces, medical depth-based applications, indoor and outdoor scenes). The most popular datasets are highlighted, together with bibliographic information (such as the number of citations). Furthermore, different aspects of the datasets are compared, common characteristics of popular datasets are described, and key recommendations for generating depth estimation datasets are suggested. The dataset description, metadata, ground truth, and relevant information i.e. (year of publication, ground truth information, size of the images, type, objects per image and number of images) are all listed in a structured way for each dataset. Also, each loss function is described in a way that can help the research community choose a right loss function for their specific tasks.

The authors hope to answer the following research questions based on the review. What are currently available datasets for the depth estimation? What are the most commonly used datasets for depth estimation and what are their distinguishing features?

How distinct are the features of such datasets and what are their pros and cons when considering them for training by machine learning (ML) algorithms? What are the most commonly used loss functions and how they influence the model performance while training the depth estimations through ML

algorithms? What are the best practices for building a depth estimation datasets?

The rest of the survey paper is organized as follows: Section 2 describes related work, primarily other studies or surveys in the field of depth estimation. The findings of a bibliometric study are provided in section 3. A comprehensive review of depth datasets is presented in Section 4. Section 5 describes common characteristics of popular datasets. Top five state-of-the-art (SoA) depth estimation methods on three most popular datasets are presented in Section 6. In section 7, popular depth estimation loss functions are studied. A brief overview, relevant research, problems, and future research prospects are presented in Section 8. A summary of the current review is offered in section 9, while sections 10 and 11 make broad recommendations for creating new datasets to achieve scientific importance and conclusion.

II. RELATED WORKS

In this section, a review of the current SoA research is provided for depth datasets. Next, an overview of available related depth estimation research and 3D reconstruction articles is presented, followed by depth from 2D, monocular, and depth from Stereo & Multi-View depth datasets.

A. DEPTH DATASETS

The procedure of maintaining 3D information of a scene using 2D information captured by cameras is referred to as depth estimation. The authors in [8] presented a detailed analysis of image-based depth estimation and 3D reconstruction. They provided details of existing systems, shortcomings, and reconstruction approaches while briefly introducing five publicly available datasets for depth estimation. However, due to several limitations, particularly hardware (e.g., sensors and optics limitations), the applicability of such datasets is questionable for future research. The authors in [9] looked at image segmentation research using deep learning with details of five public depth datasets and briefly discussed other segmentation datasets. The authors also point out sensor limitations and future research directions, but they don't explain all the relevant datasets.

While the authors in [10] presented an analysis of a method that combines ten datasets for monocular depth estimation with results on ten datasets, a description for utilizing the datasets, however, is not presented. An overview of deep-learning algorithms for monocular depth estimation using two public datasets was published in [11]; they present the significance of using NYU-v2 and KITTI datasets and argue that comprehensive testing with other datasets is required.

Three types of depth estimation datasets were chosen and described in [12] for understanding depth estimation models.

The application of deep learning algorithms with four primary depth datasets for monocular depth estimation was studied in [13]. However, some of the relevant datasets which may influence the performance were not given much importance. The authors in [14] surveyed deep learning-based

monocular depth estimation algorithms in the visible spectrum by describing a total of seven visible spectrum datasets. Some of the existing review articles [15]–[20] focusing on depth estimation either from single or multiple views, but the accessibility of those datasets is unclear.

B. DEPTH ESTIMATION RESEARCH AND 3D RECONSTRUCTION

One of the most useful intermediate representations for action in physical environments is depth information, however, activity depth estimation remains a challenging problem in computer vision. To solve it, one must exploit many, sometimes, visual cues, subtle, short-range or long-range context, along with their corresponding information. This calls for learning-based methods. Depth estimation methods have been shown in the SoA to be a potential solution to several of problems [10], [11], [15]. Accurate depth estimation approaches can help with understanding 3D scene geometry and 3D reconstruction, which is especially significant in cost-sensitive applications and use case applications [16]. A comprehensive review of 3D reconstruction research is proposed in [8], which focuses on the work that uses deep neural network-based methods to estimate the 3D shape either from single or multi-view images [21].

C. DEPTH FROM 2D, MONOCULAR IMAGES

Estimating depth information from 2D images is one of the most important problem in the field of computer vision and image processing. Depth information can be applied in 2D to 3D reconstruction, scene refocusing, scene understanding, depth-based image editing, and 3D scene conversion. The problem of monocular depth estimation is currently best tackled with convolutional neural networks due to their properties that can be used particularly in cost-sensitive applications [22]. SoA monocular depth methods have been reviewed in [11], [17], [18], [23]–[25], which focus on both non-deep learning and deep learning methods.

D. DEPTH FROM STEREO & MULTI-VIEW

Depth from stereo or multi-view can be obtained by using two or more cameras. The main idea is that triangulation and stereo matching can be used to estimate the depth, which can be utilized in various tasks such as robotic navigation, different object grasp, collision avoidance, or broadcasting and multimedia. Various methods have been studied in [2], [4], [8], [20], [26] that focus on depth estimation from both stereo and multi-view images.

III. METHODOLOGY FOR REVIEWING DEPTH DATASETS AND LOSS FUNCTIONS EMPLOYED IN LITERATURE

Utilizing the most suitable dataset for a given task is a basic assumption for the effective training and validation of any scientific method. In the domain of depth estimation research, the lack of publicly available depth estimation datasets and loss functions present challenges for researchers for their specific task or use-case.

This section aims to provide an in-depth explanation of the methodology used to search for and collect more than 40 popular datasets and loss functions which is presented in this review. The authors defined popularity based on the citation rank within the research areas and provide a detailed list of collected datasets and loss functions, as well as reviewed papers, in subsequent sections.

A. EXPLORING THE IMAGE DEPTH RELATED RESEARCH

There are numerous literature sources related to depth estimation. This study focuses on research publications that involve depth estimation tasks such as smart mobility-based road navigation, object detection, 3D reconstruction, robotics, and self-driving cars. The search methodology illustrated in Fig. 1 is adopted as to concentrate on the most relevant papers as well as leverage popular libraries and search tools such as Web of Science, Google Scholar, and IEEE Engineering online libraries.

Keywords such as “depth estimation and 3D reconstruction”, “depth datasets, databases”, “monocular and multi view depth estimation methods” were used as search criteria which helped in identifying 634 relevant journal papers. The selection of papers was based on three main factors: (i) Computer vision, engineering, deep learning, imaging technology, autonomous vehicles and robotics, 3D reconstruction, (ii) Science citation index, and (iii) English language.

B. PRIMARY STUDIES AND ASSESSMENT OF RESEARCH QUALITY

Following the research methodology (Fig. 1), the initial filter search using the datasets keyword retrieved 321 results for depth datasets and 212 results for loss functions out of 634 papers, the results were further analysed by title and abstract which filtered out 145 and 104 research articles respectively. Next, it is analysed that the text with the criteria being the selection of those articles in which the authors discussed at least one depth image datasets and loss function, carried out manually by reading the selected research articles. Such analysis helped in further reducing the number of papers to 92 and 80, which were further filtered down to the most relevant 52 and 48 articles using full-text-based selection criteria. As per the last stage’s criteria, the following categories of articles are excluded:

1. Those publications that are not directly related to depth estimation research. Examples include studies on 3D reconstruction or segmentation tasks datasets.
2. Reproductions or the same research work appearing in several places.
3. Studies that are concerned with human depth but do not make use of any depth datasets (e.g., review studies).

C. ANALYSIS OF THE MOST RELEVANT DATASETS

The methodology discovered that about 61% of the total papers in this domain considered at least one dataset in their experimental study. Additionally, 51% of the publications

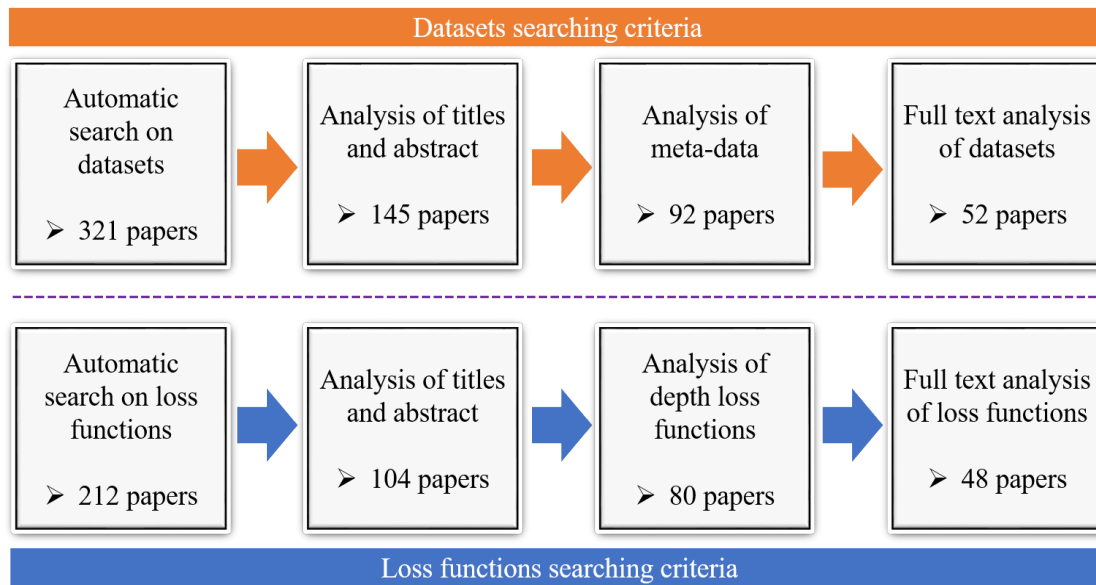


FIGURE 1. An illustration of the methodology adopted for conducting the survey categorizes depth estimation databases and the loss functions.

considered two or more than two datasets. Fig. 2 shows the results, where it is highlighted that the overall number of citations for the most popular datasets. The figure indicates that the most highly ranked depth datasets are KITTI, Cityscapes, and NYU-V2, with a citation count of 141, 94, and 78 in 120, 70, and 52 papers, respectively. This implies that about 25% of the studies considered these datasets for depth estimation tasks. These datasets are considered benchmark datasets in about 242 (77%) research studies.

The descriptions and comparisons of numerous criteria used to assist in navigating current publicly available datasets are presented by focusing on the usefulness of the datasets for specific study areas. The nature of the data imposes several restrictions on the availability of the datasets to the public. To assess the current availability of each dataset, their accessibility, in terms of access and obtaining a copy, is confirmed manually by the authors for each dataset. The test for access to each of the datasets included

checking free access and an email-based inquiry to the host institution.

IV. PUBLICLY AVAILABLE DEPTH ESTIMATION DATASETS

This section presents an overview with tabular summaries of the most widely used image depth datasets and classifies them into different use case applications.

Numerous interesting datasets are available for training depth estimation models for both multi-view and monocular images. The datasets general metadata includes details on the number of objects, scenes, and the number of RGB and depth images. The ground truth includes different types of knowledge available in each dataset, including depth, mesh, camera trajectories, video, poses, point cloud, semantic label, trajectory, and dense multi-class labels.

With the growth (evolution) in image depth estimation research, increasing efforts are made in generating larger and

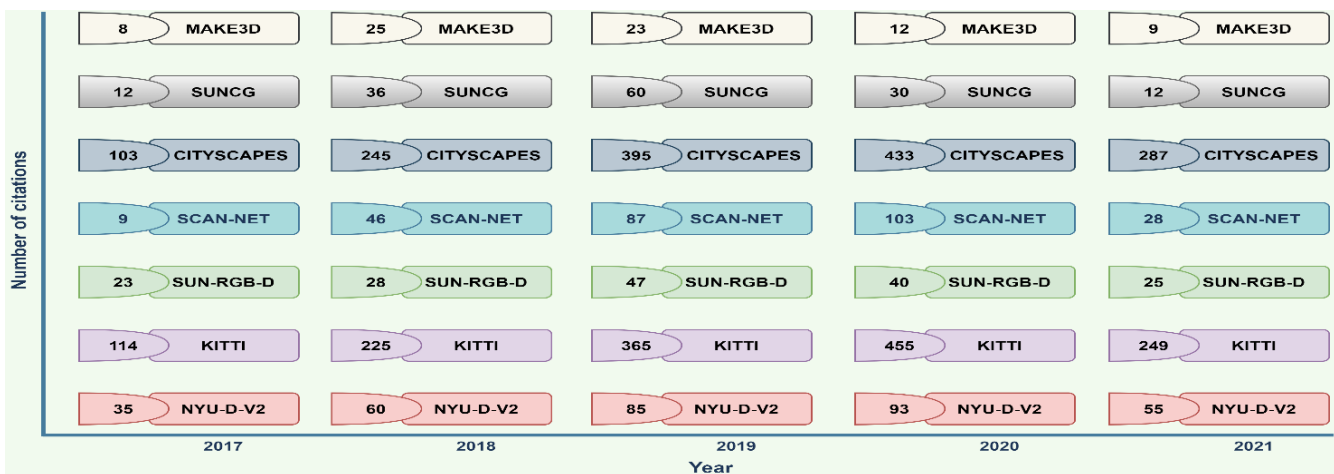


FIGURE 2. Mag an illustration of database according to the number of citations in each year from 2017 to 2021. The number against each database represents the total citations in each year.

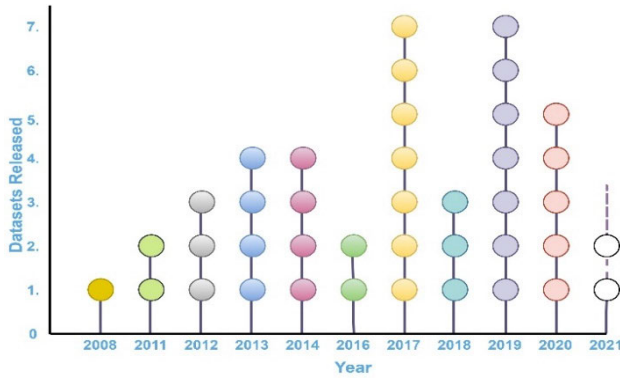


FIGURE 3. The amount of depth datasets released each year, with predicted releases in 2021 represented as a dashed line.

more ambitious depth estimation datasets. One growing trend is the increasing number of new publicly available depth estimation datasets becoming available each year over the last ten (10) years. This trend is shown in Fig. 3. A structured taxonomy showing the importance of the depth estimation datasets is given in Fig. 4. The datasets are further divided into different environments (i.e., real/synthetic indoor/outdoor, static indoor/outdoor, and real/rendered facial) in Figure 4.

Large and diverse training sets are required for depth estimation. Since obtaining pixel accurate ground-truth depth at scale in a range of circumstances is challenging, different datasets with specific characteristics and biases have been proposed.

A. THE TYPE AND REPRESENTATIONS OF DATA

There are different types (i.e., alphanumeric, text, image, video, point cloud, mesh, voxel) and representations of data such as (stereo 2D, 2.5D, 3D) that are used to analyse the scenes from different perspectives (e.g., angles).

The most up-to-date depth datasets are divided into many use case applications, such as (camera tracking, scene reconstruction, tracking, semantic, pose, video, streets, people i.e., identity recognition and faces, and medical depth-based applications, indoor and outdoor scenes). A detailed comparative analysis for various data representations is provided in Table 1.

Moreover, as some datasets contain data of various types and categories, Table 2 – 11 tabulates a comparative study for the data present in each dataset using the following labels:

- **RGB:** 2-dimensional visible light spectrum images.
- **Depth:** generic term for a map of per-pixel data containing depth-related information. A depth map describes at each pixel the distance to an object (e.g., distance from camera).
- **Video:** sequence of temporally consecutive visual readings.
- **Point cloud:** data composed of a collection of points representing a 3-dimensional shape, where each point has at least an x, y, z coordinate.
- **Mesh:** polygon-based representation of 3-dimensional shapes that directly captures topology and shape surface.
- **Scene:** data recording some environment such as a room.
- **Semantic:** labels mapping some data to a class in some ontology (e.g., human, vehicle, etc.).
- **Object:** data capturing features of objects such as shape or motion. Suitable for tasks such as object classification or tracking.
- **Camera:** data that can be used to track the camera’s geometrical features.
- **Action:** data recording subjects performing certain actions.

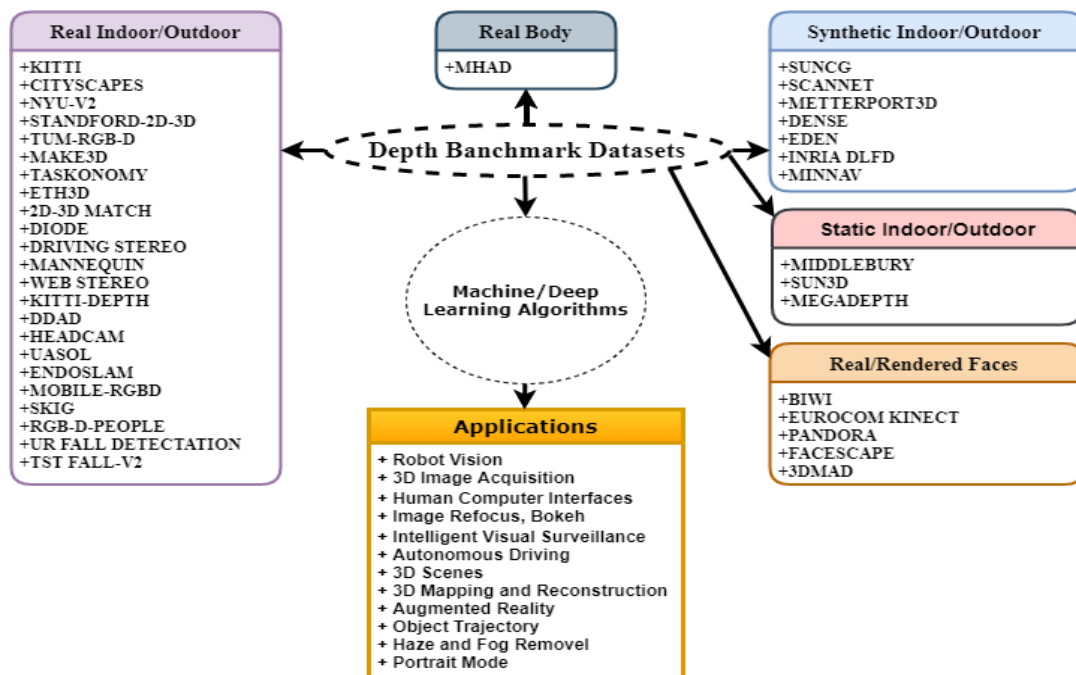


FIGURE 4. Organized classification of depth datasets studied in this paper, which shows different use case applications of each categories.

TABLE 1. Comparison between data representations.

Representation	Data Dimension	Shape Details	Memory Efficiency	Computation Efficiency
RGB	2D	⊕ ⊕ ⊕	⊕	⊕ ⊕
Depth	2.5D	⊕ ⊕ ⊕	⊕	⊕ ⊕
Mesh	3D	⊕	⊕ ⊕ ⊕	⊕ ⊕
Voxel	3D	⊕ ⊕ ⊕	⊕ ⊕	⊕ ⊕ ⊕
Point cloud	3D	⊕ ⊕	⊕ ⊕ ⊕	⊕ ⊕ ⊕
Octree	3D	⊕ ⊕ ⊕	⊕ ⊕	⊕ ⊕
TSDf	3D	⊕ ⊕	⊕ ⊕ ⊕	⊕ ⊕
Stixel	2.5D	⊕ ⊕ ⊕	⊕	⊕

⊕ : low; ⊕ ⊕ : moderate; ⊕ ⊕ ⊕ : high

- **Trajectory:** data capturing the path of motion or action being performed by some object or entity.
- **Pose:** data specifying human pose information, such as head pose.

B. DEPTH DATASETS FOR PEOPLE DETECTION AND ACTION RECOGNITION

Datasets that capture people doing different tasks like walking and acting as well as human recognition and activity depth datasets can play an important role. By employing depth map people datasets, the goal is to recognize the subject's identity, gender, or other qualities and activities.

1) RGB-D PEOPLE

The RGB-D people dataset [27] contains over 3,000 RGB and depth frames collected from three Kinect sensors mounted vertically in a university hall. The data is comprised of up-right walking and standing humans seen from various angles with various degrees of occlusion. The data is gathered in a middle position (i.e., the lobby of a large canteen) by observing people's unscripted behaviour during lunch time. The video sequences are captured at 30Hz using a set of three Kinect v1 sensors vertically joined ($130^{\circ} \times 50^{\circ}$ field of view). This capturing device is around 1.5 meters away from the ground. It ensures that the three images are captured in a synchronized and simultaneous manner while also reducing IR projector crosstalk between the sensors. To reduce sensor biases, certain background samples are taken from another building on the College campus. Occlusions between people is present in most sequences to make the data more realistic. Following the ground truth, all frames are manually annotated with bounding boxes in 2D depth image space and subject visibility position. A total of 1,088 frames, including 1,648 instances of persons, have been labelled to smooth the evaluation of individual detection systems.

2) TST FALL DETECTION V2

During the simulation of Activities of Daily Living (ADLs) and falls, the dataset [28] contains depth frames and skeleton joints collected using Microsoft Kinect v2 and acceleration samples provided by an inertial measurement unit (IMU).

The ADLs dataset is simulated for 11 young actors. The actions listed below are included in the ADL category:

the actor sits in a chair; the actor walks and grabs an object from the floor; the actor walks and grabs an object from the floor; the performer takes a walk back and forth; the actor lies down on the floor. The following actions are included in the category of fall: In the front, the actor falls to the ground and lies down; at the back, the actor falls backward and ends up lying; at the side, the actor falls to the side and ends up lying; EUpSit, the actor falls backward and ends up sitting. Each actor performed each action three times, resulting in a total of 264 sequences. The following information is provided for each sequence: Two raw acceleration streams, provided by IMUs constrained to the actor's waist and right wrist; skeleton joints in depth and skeleton space, captured by Microsoft SDK 2.0; depth frames with a resolution of 512×424 , captured by Kinect v2; timing information, timestamps of Kinect frames and acceleration samples, useful for synchronization.

3) WEB STEREO VIDEO

The web stereo video dataset can be used for depth from monocular video sequences containing a large number of non-rigid objects, such as people. To learn non-rigid scene reconstruction cues, [2] includes 553 stereoscopic videos from YouTube. This dataset contains a wide range of scene types as well as several non-rigid features.

4) MANNEQUIN CHALLENGE

In-wild recordings of people in static poses as a handheld camera pan around the environment are available in the mannequin challenge dataset [29]. The dataset is split into three parts for training, validation, and testing. The mannequin challenge is a film collection of people replicating mannequins by freezing in a variety of natural poses as a handheld camera covers the scene. More than 170K frames and associated camera postures were retrieved from around 2,000 YouTube videos in the dataset. SLAM and bundle adjustment techniques were used to calculate the camera poses. The Mannequin Challenge dataset has been used to train the model for predicting dense depth maps from common video with the camera and participants in the scene moving.

5) MHAD

Except for one senior person, the Berkeley Multimodal Human Action Database (MHAD) [30] contains 11 acts done

by 7 male and 5 female subjects between the ages of 23 and 30. All the individuals repeated each action five times, resulting in about 660 action sequences and 82 minutes of total recording time. In addition, they recorded a T-pose for each subject which can be used for the skeleton extraction; as well as the background data (i.e., with and without the chair used in some of the activities). Actions with movement in both upper and lower extremities, such as jumping in place, jumping jacks, and throwing; actions with high dynamics in upper extremities, such as waving hands and clapping hands; and actions with high dynamics in lower extremities, such as sitting down and standing up, are included in the specified set of actions. The subjects were given instructions on what action to complete before each recording, but no exact specifics on how the activity should be carried out were supplied (i.e., performance style or speed). As a result, some of the activities have been performed in a variety of styles by the individuals (e.g., punching, throwing). Depth data is collected using two Microsoft Kinect v1 sensors placed in opposite directions to prevent active pattern projection interference.

6) UR FALL DETECTION

The dataset [31] has 70 sequences (30 falls + 40 activities of daily living). Falling events are captured using two Microsoft Kinect v1 cameras and accelerometric data. Only one device (camera) and an accelerometer are used to record ADL actions. PS Move (60Hz) and x-IMU (256Hz) devices were used to collect sensor data.

7) MOBILE-RGBD

On the mobile platform, MobileRGBD is a corpus dedicated to low-level RGB-D dataset [32]. It flipped the traditional corpus recording paradigm on its head. The goal is to make ground truth annotation and record reproducibility easier in the face of speed, trajectory, and environmental changes. To portray static users in the environment, they utilized dummies that do not move between recordings. It is feasible to record the same motion multiple times to validate the impact of detecting algorithms at different speeds. This benchmark corpus is for low-level RGB-D algorithms such as 3D-SLAM,

body/skeleton tracking, and face tracking with a mobile robot. Depth data was collected using a Kinect v2 sensor.

C. DEPTH DATASETS FOR FACES AND POSES

Aside from providing a low-cost camera sensor that produces both RGB and depth information, the depth camera sensor also allows a faster human-skeletal tracking. This tracking technique can offer the exact location of human body joints across time, making analyses of complex human behaviours simpler and faster. As a result, deducing human faces from depth images or combining depth and RGB images has received much attention. In recent years, several of these new depth datasets have been developed to help in the verification of human facial activity analysis techniques.

1) BIWI

BIWI dataset [33] with over 15K images of 20 people (6 females and 14 males - 4 people were recorded twice). A depth image, the associated RGB image (both 640 × 480 pixels), and the annotation are provided for each frame. The range of head poses is approximately + – 75 degrees yaw and + – 60 degrees pitch. The ground truth is provided in the form of the head’s 3D location and rotation. Depth data is acquired using a Kinect v1 sensor.

2) EURECOM KINECT FACE

The multimodal face images of 52 persons (14 females, 38 males) acquired by Kinect v1 are included in the Dataset [34]. The data was collected in two sessions at different times (about half a month). In each session, the dataset provides the facial images of each person in 9 states of different facial expressions, lighting, and occlusion conditions: neutral face, smiling, open mouth, strong illumination, occlusion of eyes by sunglasses, occlusion of mouth by hand, occlusion of side of face by paper, right profile, and left profile. The RGB color image, the depth map (given in two forms of the bitmap depth image and the text file containing the actual depth levels sensed by Kinect), and the 3D image are all produced in three formats. The dataset also includes manual landmarks for six facial positions: left eye, right eye, the tip of the nose, left corner of the mouth, right corner of the mouth, and the chin.

TABLE 2. Depth datasets for people detection and action recognition.

DATASET	RGB	DEPTH	VIDEO	POINT-CLOUD	MESH	SCENE	SEMANTIC	OBJECT	CAMERA	ACTION	TRAJECTORY	POSE
RGB-D -P [27]	√	√	√	×	×	×	×	×	×	×	×	√
TST-F-V2[28]	√	√	×	×	×	√	×	×	×	×	×	×
W-S [2]	√	√	√	×	×	√	×	×	×	×	×	×
M-E [29]	√	√	×	√	√	√	×	×	√	×	×	√
MHAD [30]	√	√	×	√	×	√	×	×	√	×	×	√
U-F-D [31]	√	√	×	×	×	√	×	×	×	×	×	×
M-RGBD [32]	√	√	×	×	×	√	×	×	×	×	×	√

√: AVAILABLE; ×: NOT AVAILABLE; M-E: MANNEQUIN; W-S: WEB STEREO; RGB-D-P: RGB-D PEOPLE; TST-FALL-V2: TST FALL-V2; U-F-D: UR FALL DETECTION; M-RGBD: MOBILE-RGBD.

TABLE 3. Properties of depth datasets for people detection and action recognition.

No	Name	Year	Ground truth type	Size	Objects per image	Type	No images
1.	RGB-D-P [27]	2011	Depth	640 × 480	Multiple	real, in outdoor	3500
2.	MHAD [30]	2013	depth, 3D position	3600 × 3600	Multiple	real, body	36940
3.	MOBILE-RGBD [32]	2014	Depth	1900 × 1080	Multiple	real, indoor	36257
4.	UR FALL DETECTION [31]	2014	Depth	640 × 480	Multiple	real, indoor	7000
5.	TST FALL-V2 [28]	2016	Depth	640 × 480	Multiple	real, indoor	15000
6.	WEB STEREO [2]	2019	Depth, 3D models	1080 × 1080	Multiple	Real, outdoor	5000
7.	MANNEQUIN [29]	2019	Depth	640 × 480	Multiple	Real, in-outdoor	170K

3) PANDORA

The Pandora dataset [35] has 250K full-resolution RGB and depth images, obtained from a Kinect v2 sensor, as well as their annotations. For head centre localization, head pose estimation, and shoulder pose estimation, the Pandora dataset is frequently utilized.

4) FACESCAPE

The FaceScape dataset [36] contains large-scale and high-quality 3D face models, parametric models, and multi-view images. The camera settings, as well as the subjects age and gender, are all included. The information has been made available to the public for non-commercial research purposes. The FaceScape dataset contains 18,760 textured 3D faces, each with 20 distinct expressions, captured from 938 subjects. The pore-level facial geometry is also processed to be topologically uniformed in the 3D models. For rough shapes, these fine 3D facial models can be represented as a 3D morphable model, and for detailed geometry, as displacement maps. Using a deep neural network to learn the expression specific dynamic features, a novel approach is proposed that takes advantage of the large-scale and high-accuracy dataset.

5) 3DMAD

The 3D Mask Attack Database [37] (3DMAD) is a database for spoofing biometric (facial) data. It contains 76500 frames of 17 people captured with Kinect v1 for real-time spoofing attacks. A depth image (640 × 480 pixels – 1 × 11 bits), the corresponding RGB image (640 × 480 pixels – 3 × 8 bits), and carefully labelled eye positions make up each frame (concerning the RGB image). For each person, data is collected in three separate sessions such that in each session capturing five 300-frame recordings. The recordings are conducted in a controlled environment with a frontal view and neutral expression. The first two sessions are dedicated to real-world samples, in which individuals are recorded with a two-week gap between captures. A single operator captures 3D mask attacks in the third session (attacker).

D. PERCEPTION-BASED NAVIGATION DEPTH DATASETS (i.e., STREET SIGNS, ROADS)

The peripheral vision of humans enables them to observe more than just the focused objects, and their visual system is capable of immediately analysing various characteristics of the observed objects, such as distance, shape, motion, etc. But this is not the case with robots and other computer-based agents. Their vision relies upon the complex structure of hardware cameras and software with complicated mechanisms

for panoramic sight and perceiving depths. Due to the wide-screen views and blurred depth perception, robotics such as drones and self-driving cars typically lack the ability to provide valuable feedback as they navigate.

1) KITTI

KITTI [38] is one of the most often used datasets in mobile robots and self-driving cars. It contains hours of videos of traffic scenarios captured with a range of sensor modalities, including high-resolution RGB and grayscale stereo cameras, as well as a 3D laser scanner (LiDAR). The dataset itself does not contain ground truth for semantic segmentation. However, various researchers have annotated parts of the dataset manually to meet their needs. The authors in [39] created ground truth for 323 images from the road detection challenge, divided into three categories: road, vertical, and sky. The work in [40] annotated 252 (140 for training and 112 for testing) acquisitions, RGB and Velodyne LiDAR scan, from the tracking challenge for ten object categories including building, sky, road, vegetation, sidewalk, car, pedestrian, cyclist, sign/pole, and fence. The authors in [41] labelled 170 images for training and 46 images for testing (from the visual odometry challenge) with 11 classes: building, tree, sky, car, sign, road, pedestrian, fence, pole, sidewalk, and bicyclist.

2) CITYSCAPES

The Cityscapes dataset [42] is a large-scale dataset dedicated to the semantic evaluation of urban street scenes. It includes semantic, instance-based, and dense pixel annotations for 30 classes divided into eight groups (i.e., flat surfaces, humans, vehicles, constructions, objects, nature, sky, and void). Around 5,000 finely annotated images and 20,000 coarsely annotated images make up the dataset. The data was collected in 50 places for several months, during daylight hours and under favourable weather circumstances. It was originally shot on video; therefore, the frames were hand-picked to include a large number of dynamic objects, a dynamic scene layout, and a changing background. It also contains 5,000 polygonal annotations, 5,000 volume annotated images for both fine and course annotations, video frames, GPS coordinates, Ego-motion, and outside temperature data from the vehicle sensor and odometry. In terms of diversity, cityscapes are one of the most popular benchmark datasets.

3) DRIVING STEREO

DrivingStereo is a large-scale stereo dataset [43] that was created. It is hundreds of times larger than the KITTI stereo

TABLE 4. Depth datasets for faces and poses.

DATASET	RGB	DEPTH	VIDEO	POINT-CLOUD	MESH	SCENE	SEMANTIC	OBJECT	CAMERA	ACTION	TRAJECTORY	POSE
BIWI [33]	√	√	×	×	×	√	×	×	√	×	×	√
EURECOMKINECT [34]	√	√	×	×	×	√	×	×	√	×	×	√
3DMAD [37]	√	√	×	×	×	√	×	×	√	×	×	×
PANDORA [35]	√	√	×	×	×	√	×	×	√	×	×	×
FACESCAPE [36]	√	√	×	√	√	×	×	×	×	√	√	√

TABLE 5. Properties of depth datasets for faces and poses.

No	Name	Year	Ground truth type	Size	Objects per image	Type	No images
1.	BIWI [33]	2011	Depth	640 × 480	Multiple	real, faces	15K
2.	3DMAD [37]	2013	Depth	640 × 480	Multiple	real, faces	76500
3.	EURECOM KINECT [34]	2014	Depth	256 × 256	Multiple	real, faces	19274
4.	PANDORA [35]	2017	Depth	256 × 256	Multiple	real, faces	10295
5.	FACESCAPE [36]	2020	2D, 3D Landmarks, depth	4096 × 4096	Multiple	Rendered, faces	8K

dataset, with over 180k images covering a wide range of driving scenarios. A model-guided filtering technique from multi-frame LiDAR points produces high-quality disparity labels. Deep-learning models trained on the DrivingStereo dataset achieve higher generalization accuracy in real-world driving scenes than models trained on other datasets. The dataset contains left and right images along with disparity maps and depth maps. The total number of images 182188 is further divided into 174437 for training and 7751 pairs for testing.

4) KITTI-DEPTH

The depth maps from projected LiDAR point clouds were matched against the depth estimation from the stereo cameras in the KITTI-depth dataset [44]. It contains 93K depth maps with corresponding raw scene and RGB images captured with LiDAR aligned with the raw KITTI Dataset. On the benchmark server, there are 86k training images, 7k validation images, and 1k test set images. This dataset will enable the training of advanced deep learning models for the problems of depth completion and single image depth prediction.

5) UASOL

The UASOL RGB-D stereo dataset [45] has 160,902 frames captured in 33 separate scenes with between 2k and 10k frames each. The frames represent different pathways, such as sidewalks, trails, and roadways, as seen through the eyes of a pedestrian. The images were extracted from HD2K video files having a resolution of 2280 × 1282 pixels and a frame rate of 15 frames per second. Each second in the sequences has a GPS geolocation identifier, and the dataset reflects various climatological circumstances. It also involves up to four people photographing the dataset several times during the day.

6) DDAD

DDAD is a new autonomous driving dataset [25] from the Toyota Research Institute (TRI) for long-range (up to 250m)

and dense depth estimation in challenging and diverse urban environments. It includes monocular movies as well as accurate ground-truth depth (over a full 360-degree field of view) generated by high-density LiDARs placed on a fleet of self-driving automobiles driving across the United States. Scenes from cities in the United States (San Francisco, Bay Area, Cambridge, Detroit, Ann Arbor) and Japan (Tokyo, Odaiba) appear in DDAD.

7) DENSE

DENSE (Depth Estimation on Synthetic Events) [46] is a novel dataset with pixel accurate ground truth. The camera specifications are set to imitate the MVSEC event camera, which has a sensor size of 346 × 260 pixels and a horizontal field of view of 83 degrees. DENSE is divided into five training sequences, two validation sequences, and one testing sequence. Each sample is a tuple containing one RGB image, the stream of scenes between 2 subsequent images, ground truth depth, and segmentation labels. Each sequence has 1000 samples at 30 frames per second.

8) HEADCAM

This dataset [47] features panoramic video captured while riding a bike around suburban Northern Virginia with a helmet-mounted camera. The videos were used to test an unsupervised learning system for estimating depth and ego motion. The videos are saved as.mkv video files with lossless H.264 compression.

E. OBJECT AND SCENE RECOGNITION DEPTH DATASETS

Object recognition determines whether the input image contains the pre-defined object, while scene recognition labels all objects in a scene in a dense manner. With the help of object recognition methods, one can distinguish the differences between objects and determine many distortions that might occur such as different occlusions levels, illumination variations, and reflections. Combining RGB and depth information could potentially improve the robustness of the feature

methods. Several depth datasets are generated for different tasks in depth object and scene recognition.

1) NYU-D V2

NYU-D V2 [48] is mainly composed of video sequences from a variety of indoor environments captured by the Microsoft Kinect v1 RGB and depth cameras. It consists of 1,449 richly annotated pairs of aligned RGB and depth images from over 450 scenes across three cities. A class and an instance number are assigned to each object (e.g., cup1, cup2, cup3, etc.). There are also 407,024 unlabelled frames in the collection. In comparison to other datasets, this one is relatively small. This dataset was used as a benchmark for indoor depth, segmentation, and classification in the representative study work.

2) SCANNET

ScanNet [49] is an indoor RGB-D dataset that includes both 2D and 3D data at the instance level. Rather than points or objects, it is a collection of labelled voxels. ScanNet v2, the most recent version of ScanNet, has collected 1513 annotated scans with a surface coverage of over 90%. This dataset is divided into 20 classes of annotated 3D voxelized objects for the semantic segmentation challenge.

3) SUN3D

SUN3D includes [50], a large-scale RGB-D video database with 8 annotated sequences. Each frame contains a semantic segmentation of the scene’s features in conjunction with the information on the camera’s position. It is made up of 415 segments captured in 254 distinct locations across 41 different buildings. Furthermore, several locations have been photographed multiple times throughout the day. Depth acquisition was performed using the Asus Xtion Pro Live which utilizes depth from structured light technology.

4) SUN RGB-D

There are 10335 realistic RGB-D images of room scenes in the SUN RGB-D dataset [51]. Each RGB image has a depth and segmentation map that corresponds to it. There are almost 700 different objects with labelled categories. There are 5,285 and 5,050 images in the training and testing sets, respectively. The entire dataset is fully annotated, including 146,617 2D polygons and 58,657 3D bounding boxes with detailed object orientations, as well as a 3D room layout and scene categorization. This dataset allows us to train data-hungry scene-understanding algorithms, evaluate them using direct and relevant 3D metrics, minimize overfitting to a limited testing set, and investigate cross-sensor bias. Four sensors,

TABLE 6. Perception-based navigation depth datasets (i.e., street signs, roads).

DATASET	RGB	DEPTH	VIDEO	POINT-CLOUD	MESH	SCENE	SEMANTIC	OBJECT	CAMERA	ACTION	TRAJECTORY	POSE
KITTI [38]	√	√	×	√	×	√	√	√	√	×	×	×
CITYSCAPES [42]	√	√	√	×	×	√	√	√	√	√	×	×
D-S [43]	√	√	×	×	×	√	×	×	×	×	×	×
K-D [44]	√	√	×	×	×	√	√	√	×	×	×	×
UASOL [45]	√	√	×	×	×	√	√	×	√	×	×	√
DDAD [25]	√	√	×	√	√	√	×	×	×	×	×	×
DENSE [46]	√	√	√	×	×	√	√	×	×	×	×	×
H-CAM [47]	√	√	×	×	×	√	×	×	×	×	×	×

√: AVAILABLE; ×: NOT AVAILABLE, D-S: DRIVING STEREO; K-D: KITTI-DEPTH; H-CAM: HEADCAM.

TABLE 7. Properties of perception-based navigation depth datasets (i.e., street signs, roads).

No	Name	Year	Ground truth type	Size	Objects per image	Type	No images
1.	KITTI [38]	2012	Depth	1382 × 512	Multiple	Real, outdoor	12919
2.	CITYSCAPES [42]	2016	Semantic, instance-wise, depth	1024 × 2048	Multiple	Real, outdoor	25000
3.	KITTI-DEPTH [44]	2017	Depth	1382 × 512	Multiple	Real, outdoor	94K
4.	DRIVING STEREO [43]	2019	Disparity, depth	881 × 400	Multiple	Real, outdoor	182188
5.	HEADCAM [47]	2019	Depth	512 × 128	Multiple	real, outdoor	27538
6.	UASOL [45]	2019	depth, segmentation, GPS	2280 × 1282	Multiple	real, outdoor	160902
7.	DENSE [46]	2020	Depth	346 × 260	Multiple	Synthetic	8000
8.	DDAD [25]	2020	2D, 3D point cloud	1936 × 1216	Multiple	Real, outdoor	71600

leveraging three different depth technologies, were used for gathering depth data: Intel RealSense (depth-from-stereo), Kinect v1 and Asus Xtion (structured light), and Kinect v2 (Time-of-Flight).

5) MEGADEPTH

The MegaDepth dataset [52] contains 196 distinct locations reconstructed using COLMAP Structure-from-Motion/Multi-View Stereo (SfM/MVS) for single-view depth prediction. This dataset generates training data from multi-view Internet photo collections, a virtually limitless data source, using sophisticated SfM and MVS algorithms, and presents a large depth dataset named MegaDepth. Data obtained by MVS has its own set of difficulties, such as noise and unreconstructed objects. These issues are addressed by new data cleaning methods, as well as automatically enriching data with ordinal depth relations obtained by semantic segmentation.

6) DIODE

DIODE (Dense Indoor/Outdoor DEpth) [53] is the first standard dataset for monocular depth estimation that includes a variety of indoor and outdoor scenarios captured with the same hardware setup. There are 8,574 indoor and 16,884 outdoor samples in the training set, each with 20 scans. The validation set consists of 325 indoor and 446 outdoor samples obtained from ten separate scans. The indoor training and validation splits have a ground truth density of around 99.54 percent and 99.54 percent, respectively. With 67.19 percent for training and 78.33 percent for validation subsets, the density of the outdoor sets is naturally lower. The datasets ranges are 50m and 300m indoors and outdoors, respectively. Depth data is acquired using the FARO LiDAR.

7) MIDDLEBURY

The Middlebury Stereo dataset [54] contains pixel-accurate ground-truth disparity data and high-resolution stereo sequences with complicated geometry. The ground-truth disparities are obtained using a unique technique that uses structured illumination and does not require the light projectors for calibration. The Middlebury dataset, which contains 38 realistic indoor scenes taken through a structured light scanner, was one of the first datasets for stereo matching. A modified version of the Middlebury dataset with 33 new indoor scenes presented to provide a more accurate annotation at a resolution of 6 Megapixels. They are, however, generally small in size due to the difficulty and expensive cost of creating such exact and dense stereo datasets, which also leads to the problem of low variability. In an indoor setting with controlled lighting, the scenes are limited.

8) EDEN

EDEN (Enclosed garDEN) is a synthetic multimodal dataset for nature-oriented applications [55]. More than 300,000 images were captured from more than 100 garden models in the dataset. Semantic segmentation, depth, surface normals,

intrinsic colours, and optical flow are among the low/high level vision modalities labelled on each image.

9) INRIA DLFD

The INRIA Dense Light Field Dataset (DLFD) [55] is a light field dataset for testing depth estimation methods. There are 39 scenes in DLFD with a disparity range of $[-4,4]$ pixels. The light fields have a 512×512 spatial resolution and a 9×9 angular resolution.

10) SUNCG

The SUNCG dataset [56] contains 45,622 scenes with realistic room and furniture layouts that were generated manually using the Planner5D platform. Planner5D is a web-based interior design tool that lets users construct multi-floor room layouts, add furniture from a library, and arrange it in the rooms. After deleting duplicated and empty scenes, a simple Mechanical Turk cleaning operation was used to improve the data quality. During the work, the authors display a set of top view renderings of each level and ask the participants to vote on whether or not this is a valid apartment floor. They take three votes for each floor, and a floor is considered valid if it receives at least two positive votes. They have 49,884 valid floors, 404,058 rooms, and 5,697,217 object instances from 2,644 unique object meshes containing 84 categories in the end. They also manually assigned category labels to all the library items.

11) STANFORD 2D-3D

The Stanford 2D-3D dataset [49] collects mutually registered modalities from 2D, 2.5D, and 3D domains, as well as instance-level semantic and geometric annotations, across six indoor areas. It includes more than 70,000 RGB images, as well as depths, surface normals, semantic annotations, global XYZ images, and camera information. Depth data was collected using the Matterport camera, which combines 3 structured-light sensors at different pitches to capture 18 RGB and depth images during a 360° rotation at each scan location.

12) MATTERPORT3D

The Matterport3D dataset [57] is a big RGB-D dataset that can be used to analyze scenes in indoor areas. It is made up of 194,400 RGB-D images and features 10,800 panoramic views inside 90 real building-scale sceneries. Surface construction, camera postures, and semantic segmentation are all annotated in each scene, of a residential building with many rooms and floor levels. The Matterport camera is also used for this dataset.

13) TASKONOMY

Taskonomy [58] offers a vast and high-quality dataset of various indoor environments. This dataset contains comprehensive pixel-level geometry information via aligned meshes, as well as semantic information, derived from ImageNet, MS COCO, and MIT Places, camera positions, complete

camera intrinsic parameters, and high-quality images, making it three times the size of ImageNet. This is accomplished by searching a latent space for (first and higher order) transfer learning dependencies across a dictionary of twenty-six 2D, 2.5D, 3D, and semantic tasks.

14) ETH3D

ETH3D is a MVS benchmark/3D reconstruction benchmark that covers a wide range of indoor and outdoor environments [4]. A high-precision laser scanner was used to generate ground truth geometry. Images were captured using a DSLR camera and a synchronized multi-camera system with variable field-of-view. Instead of carefully constructing scenes in a controlled laboratory environment as in Middlebury, ETH3D provides the full range of challenges of real-world photogrammetric measurements. However, it still suffers from a lack of data samples and variability.

15) 2D-3D MATCH

The 2D-3D Match dataset [59] is a novel 2D-3D correspondence dataset that takes advantage of the availability of various 3D datasets from RGB-D scans. The data from SceneNet and 3DMatch are specifically utilised. There are 110 RGB-D scans in the training dataset, with 56 images from SceneNet and 54 scenes from 3DMatch. The following is how the 2D-3D correspondence data is generated. A set of 3D patches from various scanning viewpoints is extracted from a 3D point randomly sampled from a 3D point cloud. Each 3D patch's 3D position is re-projected into all RGB-D frames for which the point lies in the camera frustum, taking occlusion into consideration, to find a 2D-3D correlation. Around the re-projected point, the matching local 2D patches are extracted. Around 1.4 million 2D-3D correspondences are collected in total.

16) 3D60°

360° [60] repurposed newly released large scale 3D datasets, rendering them to 360, and creating high-quality 360 datasets with ground truth depth annotations. 3D60 is a collection of datasets created as part of multiple 360° vision research projects (Matterport-3D, Stanford 2D-3D, SunCG). It consists of multi-modal stereo representations of scenarios generated from large-scale 3D datasets, both realistic and synthetic.

17) MINNAV

MinNav is a synthetic dataset based on the sandbox game Minecraft [61]. To generate rendered image sequences with time-aligned depth maps, surface normal maps, and camera poses, the dataset employs multiple plug-in applications. Because of the big gaming community, there is an extremely large number of 3D open-world environments where players can identify acceptable shooting locations and create data sets, as well as create scenes in-game. Sildur renders 300 monocular color images for each camera trajectory, which are stored as 8-bit PNG files with lossless compression. The fps

is being adjusted from 10 to 120 and render at 800×600 with $\text{fov}=70$ and $\text{fps}=10$.

18) MAKE3D

The Make3D dataset [62] is a monocular depth estimation dataset with 400 single training RGB and depth map pairs and 134 test samples. While the RGB images have a high resolution, the depth maps have a low resolution of 305×55 generated from a custom 3D laser scanner.

19) TUM RGB-D

TUM RGB-D [63] is an RGB-D indoor dataset that contains colour and depth images from a Microsoft Kinect v1 sensor along with the sensors ground-truth trajectory. The data was captured at a full-frame rate (i.e., 30 Hz) and with a sensor resolution of 1 megapixel (i.e., 640×480). A high-accuracy motion-capture system with eight high-speed tracking cameras provided the ground-truth trajectory (i.e., 100 Hz).

F. DEPTH DATASETS FOR MEDICAL APPLICATIONS

In the last decade, medical recognition utilizing depth maps has seen significant research. As a result, depth maps-based medical methods are being employed for various applications, including monitoring of radiation in image-guided interventions to decrease surgical stuff exposure to X-rays, endoscopic surgeries for real time safety monitoring, and navigation analysis to support ultrasound procedures. Various datasets have been generated to address different medical task-based applications.

1) ENDOSLAM

The endoscopic SLAM dataset [64] (EndoSLAM) is a dataset for endoscopic video depth estimation. This includes 3D point cloud data for six porcine organs, capsule and standard endoscopy recordings, synthetically produced data, and clinically used conventional endoscope recordings of the phantom colon with computed tomography (CT) scan ground truth.

2) MVOR

The Multi-View Operating Room (MVOR) dataset [65] consists of 732 multi view frames captured by three RGB-D cameras (Asus Xtion Pro). Every frame consists of three RGB and depth images. The data was sampled from four days of recording in room at the hospital during vertebroplasty and lung biopsy. There are in total 2,926 2D key point annotations, 4,699 bounding boxes and 1,061 3D key point annotations.

3) Cholec80

The Cholec80 dataset [66] consists of 80 videos for cholecystectomy surgeries performed by different surgeons. The videos were shot at a frame rate of 25 frames per second. The timing (at 25 frames per second) and tool presence annotations are included in the dataset (at 1 fps). The dataset is divided into two equal-sized subgroups (i.e., 40 videos each). There are around 86K annotated images in the first subset.

TABLE 8. Object and scene recognition depth datasets.

DATASET	RGB	DEPTH	VIDEO	POINT-CLOUD	MESH	SCENE	SEMANTIC	OBJECT	CAMERA	ACTION	TRAJECTORY	POSE
NYU-v2[48]	√	√	√	×	×	√	√	√	√	×	×	×
SCANNET [49]	√	√	√	×	×	√	√	×	×	×	×	×
SUN3D [50]	√	√	√	×	×	×	√	×	√	×	×	×
SUN RGB-D [51]	√	√	×	×	×	√	√	×	×	×	×	×
M-D [52]	√	√	×	×	×	√	×	×	×	×	×	×
DIODE [53]	√	√	×	√	√	√	×	×	×	×	×	×
MB [54]	√	√	×	×	×	√	×	√	√	×	×	×
EDEN [55]	√	√	×	√	√	√	×	×	×	×	×	×
I-D [55]	√	√	×	×	×	√	×	√	×	×	×	×
SUNCG [56]	√	√	×	×	√	×	√	√	×	×	×	×
S-2-3D [49]	√	√	×	√	√	√	√	√	×	×	×	×
M-3D [57]	√	√	×	√	√	√	√	√	×	×	×	×
TASKONOMY [58]	√	√	√	√	√	√	√	√	×	×	×	×
ETH3D [4]	√	√	×	×	×	√	×	×	×	×	×	×
2-3D [59]	√	√	×	√	√	√	×	×	×	×	×	×
360°[60]	√	√	×	√	√	√	×	×	×	×	×	×
MINNAV [61]	√	√	×	√	√	√	×	×	√	×	×	√
TUM-D [63]	√	√	×	×	×	√	×	×	×	×	√	×
MAKE3D [62]	√	√	×	×	×	√	×	×	×	×	×	×

√: AVAILABLE; ×: NOT AVAILABLE, S-2-3D: STANFORD 2D-3D; MB: MIDDLEBURY; M-3D:METTERPORT3D; M-D: MEGADEPTH, 2-3D:2D-3D MATCH, I-D: INRIA DLF

TABLE 9. Properties of object and scene recognition depth datasets.

No	Name	Year	Ground truth type	Size	Objects per image	Type	No images
1.	MAKE3D [62]	2008	depth	640 × 480	Single	Real, outdoor	400
2.	TUM RGB-D [63]	2012	depth	640 × 480	Single	Real, indoor	1510
3.	NYU-V2[48]	2012	Dense depth	640 × 480	Multiple	Real, indoor	1449
4.	SUN3D [50]	2013	depth, semantic, 3D Gt	640 × 480	Multiple	Static, indoor	depth,3D
5.	MIDDLEBURY [54]	2014	depth	1080 × 1080	Multiple	Static, indoor	8640
6.	SUN RGB-D [51]	2015	depth, semantic, 2D and 3D	640 × 480	Multiple	Static, indoor	10335
7.	SCANNET [49]	2017	labelled voxels, depth, 2.5depth	1920 × 1080	Multiple	Synthetic, indoor	2.5M
8.	SUNCG [56]	2017	2D, 3D, volumetric, Depth	256 × 160	Multiple	Synthetic, Indoor	45622
9.	STANFORD 2D-3D [49]	2017	2.5 depth, meshes, point cloud	1080 × 1080	Multiple	Real, indoor	70496
10.	METTERPORT3D [57]	2017	depth, 2D, 3D semantic	1280 × 1024	Multiple	Synthetic, indoor	194400
11.	ETH3D [4]	2017	2D, 3D	24 Mpx	Multiple	Real, in-outdoor	1024
12.	TASKONOMY [58]	2018	2D, 3D, and semantic	1080 × 1080	Multiple	Real, indoor	4.6M
13.	3D60° [60]	2018	depth, 360	512 × 256	Multiple	Rendered	35985
14.	MEGADEPTH [52]	2018	depth	640 × 480	Multiple	Static, outdoor	1545
15.	DIODE [53]	2019	depth, Surface normal	1024 × 768	Multiple	Real, in-outdoor	25458
16.	INRIA DLF [55]	2019	depth, disparity	512 × 512	Multiple	Synthetic	1534
17.	2D-3D MATCH [59]	2020	2D, 3D	1080 × 1080	Multiple	Real, indoor	1.4M
18.	MINNAV [61]	2020	depth, surface normal, camera poses	800 × 800	Multiple	Synthetic	300
19.	EDEN [55]	2021	depth, segmentation, surface normal	1080 × 1080	Multiple	Synthetic	300K

Ten videos from this selection have also been thoroughly annotated with tool bounding boxes. The evaluation subgroup (the second subset) is utilized to put the algorithms for tool presence detection and phase recognition to the test.

4) xawAR16

The xawAR16 dataset [67] is multi-view RGB-D camera dataset that was created in an operating room (IHU Strasbourg) to test the tracking and relocalization of a hand-held

moving camera. To create such a dataset, three RGB-D cameras (Asus Xtion Pro Live) were employed. Two of them are fixed to the ceiling in such a way that they may capture views from both sides of the operating table. A third is attached to a display that is moved around the room by a user. A moving camera is fitted with a reflecting passive marker, and its ground-truth pose is determined using a real-time optical 3D measuring system. The dataset consists of 16 time-synchronized color and depth images in full sensor resolution (640×480) captured at 25 frames per second, as well as ground-truth positions of the moving camera measured at 30 frames per second by the tracking device. Each sequence includes occlusions, motion in the scene, and sudden perspective shifts, as well as varied scene layouts and camera movements.

G. EXPLANATION AND DATASETS COMPARISON

This section demonstrate brief comparison of depth datasets from several aspects. For an easy access, all the datasets are ordered by year; table 6 shows some features including the name of the datasets, the year of creation, ground truth type, size, objects per image in the dataset, type, and number of images. In terms of popularity of the datasets, the authors ranked the datasets based on the number of citations. The datasets that are available freely and with longer history always have more citations than the newer ones. Particularly Kitti, Cityscapes, Nyu-v2, Sun-RGB-D, Make3D, SceneNet, SunCG all have high number of citations compared to the rest of the datasets. However, it does not necessarily mean that the old datasets are better than the new ones. In terms of the baseline evaluation datasets for depth estimation, Kitti, Cityscapes, Nyu-v2 are the commonly used benchmarks. The depth datasets are divided into different categories of intended applications and studied properties. However, each dataset may not be limited to one specific application only (e.g. Kitti can be used for both depth and 3D reconstruction, Nyu-v2 can be used for both depth and segmentation). The data modalities include RGB, depth, indoor, outdoor, real, synthetic, semantic, labeled voxels, 3D, volumetric, meshes, point cloud, 3D landmarks, surface normals, camera poses, and segmentation. This is helpful for researchers to quickly identify the datasets of interest especially when they are working on multi-modal fusion. A link to each dataset is also provided, which can help research involved in similar studies. It is important to keep in mind that some datasets are updated while others' websites may change.

H. MIXING DATASETS FOR TRAINING ON DIVERSE DATA

To the author's knowledge, the systematic combination of many data sources has only been briefly studied. Reference [68] described a model for estimating two-view structure and motion, which they trained on a combination of smaller datasets with static scenes; although, they did not explain the impact of the method used. Reference [69] proposed a method of naively mixing datasets for monocular depth estimation with known camera parameters. Combining different datasets

can be challenge as the ground truth data is in different forms (i.e., absolute form: laser based or stereo camera with unknown camera parameters, depth from unknown scale, disparity maps) in every dataset (see table 3). A methodology that can be compatible with all ground truth representations for training deep networks is required. Furthermore, an appropriate loss function can be designed, which must be flexible and compatible with different kind of ground data sources.

Three key issues are identified by [10] and studied in detail.

- Direct vs. inverse depth representations are inherently different representations of depth.
- Scale ambiguity: depth with unknown scale (or camera parameters, camera calibration) in some data sources.
- Uncertainty about shift: some datasets only include disparity maps up to a certain known scale.

Although a stochastic optimization computation, loss function and prediction space allow for the mixing of different data sources, while it is not instantly obvious in what percentages different datasets will be merged through training.

When it comes to mixing datasets, there are two crucial approaches to consider.

1. In each minibatch, the first technique is to combine different data sources into equal parts which sample F/K training data from each dataset for a minibatch of size F , where K specifies the number of different datasets. This technique ensures that all datasets, regardless of the size, are characterized equally in the effective training set for training deep networks.
2. The second approach takes a more principled style, adapting a recent Pareto-optimal multi-task learning method [70]. They examine every dataset as a different task and try to find an approximated Pareto optimum across all datasets (i.e., a technique in which the loss on each training set cannot be reduced without raising it on at least one of the others). To minimize the multi-objective optimization criteria, it utilizes the algorithm provided in [70] that can be used for mixing different kind of ground truth data into an effective way for various tasks in computer vision-based applications.

$$\min_f (L_1(f), \dots, L_l(f))^t$$

where parameters of the model f are shared across different datasets.

V. COMMON CHARACTERISTICS OF WELL-KNOWN DATASETS

It was observed that, of the datasets mentioned above, the depth estimation datasets with the highest potentials displayed five common qualities:

- Longevity -This study finds that the datasets that were available for a longer period of time gained more attention and popularity. The KITTI is the most discussed dataset and has been accessible since its launch in 2012. It is the most frequently cited benchmark dataset despite several constraints, such as small scale. The KITTI dataset has become a standard

TABLE 10. Depth datasets for medical applications.

DATASET	RGB	DEPTH	VIDEO	POINT-CLOUD	MESH	SCENE	SEMANTIC	OBJECT	CAMERA	ACTION	TRAJECTORY	POSE
ENDOSLAM [64]	√	√	√	√	X	√	X	X	X	X	X	X
MVOR [65]	√	√	X	X	X	√	X	X	√	√	X	X
CHOLEC80 [66]	√	√	√	X	X	√	X	X	X	X	X	X
XAWAR16 [67]	√	√	X	X	X	√	X	X	√	X	X	X

TABLE 11. Properties of depth datasets for medical applications.

No	Name	Year	Ground truth type	Size	Objects per image	Type	No images
1.	CHOLEC80 [66]	2017	depth, 3D	640 × 480	Multiple	real	86K
2.	MVOR [65]	2018	depth, 3D	640 × 480	Multiple	real	8357
3.	XAWAR16 [67]	2021	depth, 3D	640 × 480	Multiple	real	64754
4.	ENDOSLAM [64]	2021	depth, 3D	1350 × 1080	Multiple	real	64587

benchmark for comparing new results and methods for depth estimation and 3D reconstruction tasks.

- Scale – The number of samples and subjects in a dataset plays a critical role in its popularity. A dataset must have enough sample data features for successful statistical research. Datasets with many samples (and thus a higher statistical relevance) provide objective standards. In conjunction with the dataset size, some other features such as the methodology of its representation are also important.

- Timing – It is observed that the most popular datasets provided novel features and facilitated research that was not possible with previously available public datasets. The KITTI dataset, which was the first publicly available depth outdoor dataset, the NYU-V2 dataset, which was the first dataset to add indoor imaging, and the Cityscapes dataset, which was the first to feature high-resolution images, are all good examples.

- Data quality - The data quality plays a critical role in providing the information about its use in the given situation (e.g., data analysis). It is worth noting that the datasets with details for information collection usually get more attention than the rest of the datasets (e.g., NYU-D V2, FaceScape, Cityscapes).

- The Right Data Transformation - Once generated, the datasets are modified for meeting particular performance objectives while using the machine learning algorithms. Domain knowledge and algorithm features/functions can help determine the best type of transformation to increase the training performance. Datasets that include tools for cleaning, transforming, and preparing data for training are popular than research-oriented datasets.

VI. STATE-OF-THE-ART DEPTH ESTIMATION METHODS ON THREE MOST POPULAR DATASETS

The performance of the top five SoA algorithms on popular depth estimation benchmarks is tabulated in this section. It’s worth noting that, while most deep networks report their results using standard datasets and metrics, some don’t, making it impossible to compare SoA methods across the

board. Furthermore, only a small percentage of papers provide reliable additional information, such as execution time and memory footprint, which is critical for industrial depth estimation model applications (such as drones, self-driving cars, robotics, and so on) that must run on embedded consumer devices with limited processing power and storage and thus require efficient, lightweight models. The performance of the top five SoA deep learning-based depth estimation models on three of the most popular datasets is summarized in Tables 12-14. 3d-ken-burns [71] is the best of the other methods trained on the NYU-V2 dataset, while AdaBins [72] is better on the KITTI dataset and HRNetV2 [79] is better on the cityscapes dataset.

VII. AN OVERVIEW OF LOSS FUNTIIONS FOR DEPTH ESTIMATION

Deep learning-based methods usually optimize a regression model on the reference depth map. For depth regression tasks, defining an appropriate loss function is the main challenge faced by the SoA methods. Optimisation algorithms are used by neural networks (i.e., stochastic gradient descent to minimize the errors in the algorithm). The loss function, which measures how well or poorly the model performs, is used to calculate this error. There are several noteworthy loss functions that have been employed in depth estimation problems where deep neural networks are used to forecast depth maps from a single or multiple images.

A. LEAST SQUARE LOSS

To supervise the training process of the models, the differences between the real depth y and predicted \check{y} maps are used. For the depth values, the L_2 loss function [73] can be represented as (L_2) and is defined as:

$$L_2(y, \check{y}) = \frac{1}{N} \sum_i^N (y_i - \check{y}_i)^2 \tag{1}$$

As a result, depth estimation architectures predict the ground truth to learn the depth information of the scenes.

TABLE 12. Results of top five SoA depth estimation models on the NYU-V2 dataset.

Method	Dataset	RMS	Year
3d-ken-burns [71]	NUY-V2	0.305	2019
AdaBins [72]	NUY-V2	0.364	2020
TransDepth [73]	NUY-V2	0.365	2021
BTS [74]	NUY-V2	0.407	2019
Optimized, freeform [75]	NUY-V2	0.432	2019

TABLE 13. Results of top five SoA depth estimation models on the KITTI Eigen split dataset.

Method	Dataset	AbsRel	Year
AdaBins [72]	KITTI Eigen	0.058	2020
LapDepth [76]	KITTI Eigen	0.059	2021
DPT-Hybrid [77]	KITTI Eigen	0.062	2021
BTS [74]	KITTI Eigen	0.064	2019
DORN [78]	KITTI Eigen	0.072	2018

B. SCALE-INVARIANT LOSS

During the training stage, depth estimation approaches use the ground truth of depth y and the corresponding model predicts the log depth. The training Scale-invariant loss function [73] (L_{SI}) can be represented by (L_{SI}) for the depth values and is defined as:

$$L_{SI}(y, \check{y}) = \frac{1}{N} \sum_i^N (\log(y_i) - \log(\check{y}_i))^2 - \frac{\lambda}{N} \left(\sum_i^N \log(y_i) - \log(\check{y}_i) \right)^2 \quad (2)$$

λ refers to the balance factor and is set to 0.5.

C. BERHU LOSS

To account for data that contains outliers or heavy-tailed errors, the Ordinary Least Square (OLS) estimator is deemed ineffective in this scenario. In the case of Gaussian noise, however, Berhu loss is designed to keep good qualities. Furthermore, the adaptive Berhu penalty encourages a grouping effect, which develops one group with the highest coefficients. Berhu loss function [74] (L_{Berhu}) can be represented by (L_{Berhu}) for the depth values and is defined as:

$$L_{Berhu}(y, \check{y}) = \begin{cases} (y_i - \check{y}_i) & \text{if } (y_i - \check{y}_i) \leq c, \\ \frac{(y_i - \check{y}_i)^2 + c^2}{2c} & \text{if } (y_i - \check{y}_i) > c, \end{cases} \quad (3)$$

D. HUBER LOSS

It is known that Mean Square Error (MSE) is better for learning outliers in a dataset, but Mean Absolute Error (MAE) is better for ignoring them. However, data that appears to be outliers should not be considered in some circumstances, and those points should not be given great attention. For this reason, Huber loss function [74] (L_{Huber}) can be represented by (L_{Huber}) for the depth values and is defined as:

$$L_{Huber}(y, \check{y}) = \begin{cases} (y_i - \check{y}_i) & \text{if } (y_i - \check{y}_i) \geq c, \\ \frac{(y_i - \check{y}_i)^2 + c^2}{2c} & \text{if } (y_i - \check{y}_i) < c, \end{cases} \quad (4)$$

TABLE 14. Results of top five SoA depth estimation models on the cityscapes dataset.

Method	Dataset	Mean IoU(%)	Year
HRNetV2 [79]	Cityscapes	85.1	2020
HRNetV22 [80]	Cityscapes	84.5	2019
EfficientPS [81]	Cityscapes	84.21	2020
Panoptic-DeepLab [82]	Cityscapes	84.2	2019
DCNAS [83]	Cityscapes	83.6	2019

E. SILOG LOSS

Correctly scaling the range of the loss function can increase convergence and training outputs, while increasing the λ forces more focus on minimizing the error variance, resulting in Silog loss function. Reference [74] (L_{silog}) can be represented by (L_{silog}) for the depth values, $\lambda = 0.5$ and N represent ground truth values (i.e., the number of pixels).

By rewriting equation. 2:

$$L_{silog}(y, \check{y}) = \frac{1}{N} \sum_i^N (\log(y_i) - \log(\check{y}_i)) - \frac{1}{N} \sum_i^N (y_i - \check{y}_i)^2 + (1 - \lambda) \frac{1}{N} \sum_i^N (y_i - \check{y}_i)^2$$

In log space, variance and weighted squared mean errors is combined define the Silog loss:

$$L_{silog}(y, \check{y}) = \alpha \sqrt{L_{silog}(y, \check{y})} \quad (5)$$

F. COMMON DEPTH LOSS

Let y be a ground-truth depth map and \check{y} be its estimated depth. The common depth loss [84] L_1 is given by the entry-wise L_1 -norm for a matrix

$$L_1(y, \check{y}) = \frac{1}{HW} (y_i - \check{y}_i)_1 \quad (6)$$

where W and H are the width and height of the depth maps.

G. GLOBAL MEAN REMOVED LOSS

The global mean removed loss [84] is defined as

$$L_{GMR}(y, \check{y}) = \frac{1}{HW} ((y_i - \bar{y}_i) - (\check{y}_i - \bar{\check{y}}_i))_1 \quad (7)$$

where W and H are the width and height of the depth maps, \bar{y}_i and $\bar{\check{y}}_i$ are the average depths in y and \check{y}_i , respectively. This loss is based on the observation that, while estimating the global depth scale (i.e., average depth) from an image is unclear, predicting the relative depth of each pixel in relation to the average depth is more reliable. In some situations, such as age estimation, relative estimation is easier than absolute estimation.

H. LOCAL MEAN REMOVED LOSS

A local mean removed loss [84] L_{MR} , which penalizes the relative depth errors with respect to local $n \times n$ square regions and defined as follows:

$$L_{MR}(y, \check{y}) = \frac{1}{HW} ((y_i - y_i \oplus \frac{J_m}{m^2}) - (\check{y}_i - \check{y}_i \oplus \frac{J_m}{m^2}))_1 \quad (8)$$

where \oplus denotes the convolution, and J_m is the $n \times n$ matrix composed of all ones.

I. SSIM LOSS

The perceptual difference between two comparable images is measured using SSIM. It can't tell which of the two is superior because it doesn't know which is the "original" and which has undergone further processing like data compression. The loss function for the structural similarity index measure (SSIM) is represented by (L_{SSIM}) and can be defined as:

$$L_{SSIM}(y, \check{y}) = \left(\frac{1 - L_{SSIM}(y, \check{y})}{MaxDepth} \right) \quad (9)$$

J. PHOTOMETRIC LOSS

A SSIM term is combined with the L_1 reprojection loss due to its better performance in complex illumination scenarios. Thus, the (L_P) photometric loss [85] of the N scale is modified as

$$L_P(y, \check{y}) = \sum_i^N (1 - \lambda)(y_i - \check{y}_i)_1 + \lambda \frac{1 - L_{SSIM}(y, \check{y})}{2} \quad (10)$$

K. PRE-PIXEL SMOOTHNESS LOSS

A per-pixel smoothness loss is introduced to combine with the L_{SL} reprojection loss to encourage the inverse depth prediction to be locally smooth, as depth discontinuities often occur at image gradients. Thus, the (L_{SL}) loss is defined as

$$L_{SL}(y, \check{y}) = \sum_i^N \partial_x dte^{-\partial_x(y, \check{y})} + \partial_y dte^{-\partial_y(y, \check{y})} \quad (11)$$

L. RECONSTRUCTION LOSS

The network calculates disparity during training, and the bilinear sample is used to generate the input image, which is then used to reconstruct another image using the disparity map. The bilinear sampler is fully differentiable at the local level and smoothly integrates into a fully convolutional architecture. A L_{Huber} and SSIM is combined as a photometric image reconstruction loss, which computes the inconsistency between the input image and the reconstructed image, it is defined as follows

$$L_R(y, \check{y}) = \frac{1}{N} \sum_i^N \frac{1 - L_{SSIM}(y, \check{y})}{2} + (1 - \alpha)L_{Huber}(y, \check{y}) \quad (12)$$

M. PRIOR RECONSTRUCTION LOSS

It is consequently shown that constraining a cost function involving a polarimetry-specific geometry is valid. Furthermore, because it is dependent on both the input and output of the processing pipeline, this minimization strategy can be used to optimize a deep learning model. This method is consistent in unusual circumstances, implying a limited camera calibration or a specific azimuth to angle of polarization

thought processes. As a result, a new method provides an alternative but comparable strategy that allows for standard calibration and the release of constraints via a generalized loss term defined as follows

$$L_{PR}(y, \check{y}) = \mu minL_R + \nu \partial_x^2 dte^{-\partial_x^2(y, \check{y})} + \partial_y^2 dte^{-\partial_y^2(y, \check{y})} \quad (13)$$

N-1. SCALE INVARIANT LOSS

The scale-invariant loss [32] for a single sample is defined as

$$L_{SI}(y, \check{y}) = \frac{1}{N} \sum_i^N \rho^2(y, \check{y}) - \frac{\lambda}{n^2} \left(\sum_i^N \rho(y, \check{y}) \right) \quad (14-1)$$

where ρ function defines the scale invariant loss and $\lambda \in [0, 1]$.

N. SCALE SHIFT INVARIANT LOSS

The scale-shift-invariant loss for a single sample is defined as

$$L_{SSI}(y, \check{y}) = \frac{1}{2N} \sum_i^N \rho(y, \check{y}) \quad (14)$$

where ρ function defines the scale invariant loss.

O. POINT-WISE LOSS

Point-wise loss function (L_{depth}) can be represented by (L_1) for the depth values and is defined as:

$$L_{depth}(y, \check{y}) = \frac{1}{n} \sum_i (y_i - \check{y}_i) \quad (15)$$

P. GRADIENT LOSS

To capture the local structural consistency, a gradient loss function (L_{grad}) is proposed and can be represented by (L_{grad}), which penalize the gradient of depth around the edges of the image and can be defined as

$$L_{grad}(y, \check{y}) = \frac{1}{n} \sum_i^N y_x(e_i) + \check{y}_y(e_i) \quad (16)$$

where $y_x(e_i)$ and $\check{y}_y(e_i)$ represent the spatial derivatives of the difference between the ground truth and predicted depth for the p^{th} pixels e_i which stands ($\|y_i - \check{y}_i\|$) for the x, y-axis.

Q. SURFACE NORMAL LOSS

The surface normal loss function (L_{SN}) can be utilized to avoid minor errors and predicts the normal and estimated depth maps. The ground-truth surface norms and predicted depth are represented by

$$n_i^y = (\Psi[-\nabla_x(y_i), -\nabla_y(y_i), 1]^T)$$

and

$$n\check{y}_i = (\Psi[-\nabla_x(\check{y}_i), -\nabla_y(\check{y}_i), 1]^T)$$

The loss is calculated as the difference between the two surfaces normals, which may be expressed

mathematically as follows

$$L_{SN} = \frac{1}{n} \sum_i^n \left(1 - \frac{\langle n_i^y, n_i^{\check{y}} \rangle}{(\|n_i^y\| \cdot \|n_i^{\check{y}}\|)}\right) \quad (17)$$

where $\langle n_i^y, n_i^{\check{y}} \rangle$ denotes the inner product of the vectors.

R. PERCEPTUAL LOSS

The ability of the MSE function to capture perceptually relevant differences (such as high texture details). It is very limited in the use cases because they are defined based on differences in image pixels, minimizing the pixel averages. Therefore, a perceptual loss function is introduced to make the two more perceptible similarities by comparing feature maps between original view and reconstructed view. Denote by α the feature map obtained after the j -th convolution (after activation) of the i -th convolutional layer in the VGG-16 network and the perceptual loss is defined as the Euclidean distance between the feature maps of the original view y and the reconstructed view \check{y}

$$L_{PRL}(y, \check{y}) = \frac{1}{HW} \sum_i^N (\alpha(y_i) - \alpha(\check{y}_i))^2 \quad (18)$$

The size of the generated feature map for a specific layer in the VGG network is described by H and W . Perceptual loss, rather than pixel-by-pixel loss, is more reflective of semantic similarity between images during training. By adding perceptual loss training, the depth map generated by the model has more precise details and edge information.

S. STRUCTURE GUIDED RANKING LOSS

Structure-Guided Ranking Loss is a pair-wise ranking loss that is very broad, allowing it to be applied to a wide range of depth and pseudo-depth data. The sampling method for certain point pairs, on the other hand, might have a significant impact on the reconstruction quality. Rather than utilizing random sampling, the proposed segment-guided sampling technique and purpose is to direct the networks attention to the regions that matter most, i.e., the scene's salient depth structures, and can be characterized as

$$L_{SGL}(y, \check{y}) = \frac{1}{N} \sum_i^N (\alpha(y_i - \check{y}_i)) + L_{grad}(y, \check{y}) \quad (19)$$

T. CHAMFER LOSS

The chamfer distance between two points can be defined is

$$D(X_1, X_2) = \sum_{x \in X_1} \min_{y \in X_2} \|x - y\|^2 + \sum_{y \in X_2} \min_{x \in X_1} \|x - y\|^2$$

for a distance d between subsets in R^2 , Then the Chamfer loss function takes the form

$$L_{CL}(y, \check{y}) = \sum_i^N d(y_i - \check{y}_i) \quad (20)$$

where i indexes training samples.

U. BIN CENTER DENSITY LOSS

Bin centre density loss function can be used to follow the distribution of the depth pixels in the ground truth, and it can be defined as the set of bin centres $c(b)$ and a set of the ground truth pixels in the image X along with bi-directional Chamfer loss as a regularizes

$$L_{BCDL} = \sum_{x \in X} \min_{y \in c(b)} \|x - y\|^2 + \sum_{y \in c(b)} \min_{x \in X} \|x - y\|^2 \quad (21)$$

V. GRADIENT MATCHING LOSS

To encourage the network to output a depth map with sharp edges, gradient matching loss is used and defined as

$$L_{GML}(y, \check{y}) = \frac{1}{K} \sum_{k=1}^N \sum_{i=1}^K \left(\left| \frac{\partial}{\partial x} E \right| + \left| \frac{\partial}{\partial y} E \right| \right) \quad (22)$$

where $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ are the gradient of the prediction.

W. PAIRWISE DISTILLATION LOSS

The pairwise distillation loss is obtained in two steps. First, affinity maps for the feature maps are generated. Then the MSE between the affinity maps of the obtained features is then computed.

$$L_{PDL}(y, \check{y}) = \frac{1}{x \times y} \sum_i \sum_j (p_{ij}^t - p_{ij}^u) \quad (23)$$

where p_{ij}^t and p_{ij}^u are the affinity maps.

VIII. DISCUSSION

Over the previous two decades, available depth estimation datasets have improved, yet there are still problems to be solved. The most significant limitation is their availability, which implies that many of the datasets are only available for a limited duration. It's also worth noting that in some circumstances, when the authors prefer to give the dataset based on the asking institutions, limited access is noticed (institutions with a lower profile might typically have more problems obtaining a dataset). This negatively impacts individual researchers' ability to replicate the analysis, as well as future researchers' capabilities to publish findings derived from such datasets. The impact of aging has been studied using public datasets collected in the previous few years. Long and complex depth estimation is limited by the difficulties of following up on a large group of people over a long period of time.

The new data privacy standards, which secure personal rights, have created a relatively new challenge. In Europe, for example, the General Data Protection Regulation (GDPR) includes a right to erasure (often known as the right to be forgotten), which gives subjects the option to withdraw their consent to the use of their data and have subject-related material removed from datasets (if possible). Because of the nature of biometric data, the subject can be uniquely identified. As a result, potential changes in datasets could compromise

the determination and uniformity of reported data over time. Similar legislations are being discussed globally as a result of recent difficulties relating to the lack of realistic data. Imperfections in the mentioned collection setup and technique are also significant limitations of the current datasets. Some of the dataset generation criteria are not available, but they may be useful so that others can greatly expand the datasets possible applications. Also, the optical system information is sometimes not completely defined as well as some of the datasets lack of sensor information, capture distance, range of spectrum in the generated images, and environmental validation. Some of the datasets only provide cropped image regions of the complete scene, so information like aperture, speed shutter, and sensitivity is lacking. When collecting with mobile devices, data from the IMU (i.e., an accelerometer and a gyroscope) may be beneficial in reducing the negative effects of the rolling shutter and recognizing motion blur (e.g., smartphones). In addition, several datasets only provide compressed images, reducing the quantity of data captured by the sensor.

Due to the differences in image quality, researchers require a complete explanation of the method and capture information in different research areas. Despite the common features in research problems, smartphone depth capture research focuses on using additional sensor information available in mobile platforms (IMU or multiple imaging sensors) and computational methods to process captured images, whereas depth in motion research focuses on novel sensors and optical systems.

Many research papers underline the absence of datasets suited for evaluating a specific parameter (i.e., a constrained environment with only one parameter's variability), which leaves research conclusions and underlying reasons unclear, underlining the need for more research. In some cases, having a clear protocol description may be enough to solve the problem. If the camera specifications (usually removed for privacy concerns) were contained in the EXIF/metadata, several of these issues may be avoided. This information is generally missing from datasets created using custom-built cameras, as well as a protocol description. While many details of specialized hardware are hidden from users of other datasets, publicly accessible cameras provide such attributes by default in the image file.

There is also a mismatch between datasets acquired under visible light. In some cases, the authors used a monochromatic sensor with a band-pass filter to catch the entire visible band of light, while in others, they used mass market cameras to collect visible light in three spectral bands (separately for the colors red, green, and blue). Because the spectral sensitivity of the visible light filter differs from that of the individual color filters (even when the color bands are combined), they should not be compared. Additionally, most consumer color cameras have a Bayer filter that restricts individual band resolution to one-quarter for red and blue spectra and one-half for green; as a result, two-thirds of the color information are estimated rather than measured.

The review also found that synthetic image datasets have not got momentum in depth estimation research. Researchers prefer standard datasets (real) instead of synthetic images, despite the fact that synthetic images have a higher number of samples. The authors feel that these datasets lack the realism of research effects that occur in less confined circumstances.

Only a small percentage of distance depth capture research has focused on computational depth capture, such as using super-resolution, whereas the majority has focused on constructing a standard optical system with mirrors for the capture.

A. RELATED RESEARCH

This has been a review of existing datasets generated for performance evaluation, with a focus on depth. The datasets investigated in this work could be useful in other fields of research that use images of the human body, faces, poses, objects, indoor/outdoor, medical information, and environments.

Face tracking and segmentation have been used in a wide range of applications, from human-computer interaction to medical diagnosis. These applications usually have other well-known datasets, but they primarily share initial depth image processing, such as depth localization and segmentation. As a result, depth estimation datasets could be useful as a secondary data source. Furthermore, a useful medical diagnostic for detecting neurotransmitter and neuronal activity levels has been proven using the pupil [66]. Object recognition and classification algorithms are a comparable, but more sophisticated academic area. However, depth estimation is often a more difficult challenge. It's been utilized in medical applications, such as diagnosing computer vision syndrome and facial recognition technologies.

Biometrics datasets are restricted in that they do not contain identification information, that restricts the use of many datasets. Alternatively, unsupervised methods can play an important role in depth-based recognition problems.

B. CHALLENGES AND COMPETITIONS

An independent evaluation and standard compression analysis can greatly help current depth estimation methods in a range of applications and tasks in computer vision research. There is a well-defined baseline for the SoA methods, but the results are greatly diverse due to the datasets, training, evaluation, and implementation methodologies. These variations make it difficult to compare the methods objectively for a specific problem related to depth estimation. Many of these issues can be avoided by creating benchmark datasets and conducting independent evaluations. This ensures an objective comparison of methods by using standardized protocols and environments. Competitions and/or challenges are commonly used to organize such evaluations. This strategy stimulates competition among academics in addition to the production of publicly available datasets with uniform measurements.

C. FUTURE RESEARCH DIRECTIONS

Image-based depth estimation using deep learning approaches has shown promising results following detailed research over the last few years. However, the subject is still in its early stages, and more developments are to be expected. In this section, the authors will go over some of the hot topics right now and point out in the right direction for future research.

- **Data for training purposes is a problem:** The availability of training data is critical to the effectiveness of deep learning algorithms. Unfortunately, compared to the training datasets used in tasks like classification and recognition, the size of publicly available datasets that comprise both images and their ground truth depth is small. Due to a lack of 3D training data, 2D supervision techniques have been utilized. However, many of them rely on silhouette-based supervision and can only reconstruct the visual hull as a result. Consequently, one can expect to see more papers in the future proposing new largescale datasets with diverse environments, new weakly-supervised and unsupervised methods that leverage various visual cues, and new domain adaptation techniques in which networks trained on data from a specific domain, such as synthetically rendered images, are adapted to a new domain, such as in-the-wild images, with very little retraining and supervision. Research into realistic rendering approaches that can bridge the gap between actual and synthetically created images has the potential to help with the training data problem.
- **Generalization to unseen objects:** Most SoA studies, such as BTS and AdaBins, divide a dataset into three subsets for training, validation, and testing, and then report on the performance on the test subsets. However, it is unclear how these approaches would perform on categories of objects/images that have never been seen before. In reality, the ultimate goal of the depth estimation method is to be able to recreate any 3D shape from any set of images. Learning-based strategies, on the other hand, only work on images and objects that are part of the training set. A number of recent publications have attempted to examine this topic. However, combining classical and learning-based strategies to improve the generalization of the latter methods would be an interesting direction for future research.
- **Fine-scale depth estimation:** The coarse depth structure of shapes can be recovered using current SoA approaches. Although subsequent work has enhanced the resolution of the reconstruction by employing refinement modules, thin and small portions such as plants, hair, eyes, and fur remain unrecoverable.
- **Reconstruction versus recognition:** The difficulty of obtaining depth from images is ill-posed. As a result, effective solutions must incorporate low-level image cues, structural knowledge, and a high-level understanding of the object. Deep learning-based depth estimation algorithms are biased towards recognition and retrieval,

according to a recent study [8]. As a result, many of them have difficulty generalizing and recovering fine-scale features. Therefore, it is expected that this area of research might see more exploration in the future on how to mix top-down (i.e., recognition, classification, and retrieval) and bottom-up approaches (i.e., pixel-level reconstruction based on geometric and photometric cues). This has the potential to improve the approaches' generalization capabilities (see item (2) above).

- **Handling multiple objects in the presence of occlusions and cluttered backgrounds:** Most of the SoA approaches deal with single-object images. Images taken in the wild, on the other hand, often feature a variety of things from several categories. Detection and reconstruction within regions of interest have been used in previous studies. The modules for detection, depth, and reconstruction are all independent of one another. These tasks, however, are interrelated and might benefit from one other if completed together. Two major concerns must be solved in order to achieve this goal. The first is a lack of multiple-object reconstruction training data. Second, especially for methods that are learned without 3D supervision, creating proper CNN architectures, loss functions, and learning procedures is critical. In general, these employ silhouette-based loss functions, which necessitate precise object segmentation.
- **Data Imbalance:** Some class representations are limited in some scene understanding tasks, such as semantic labelling, whereas others have a lot of examples. Learning a model that respects both types of categories and performs equally well on frequent and less frequent ones is a challenge that requires more research.

Deep-learning algorithms for depth estimation rely largely on training datasets annotated with ground truth labels, which are difficult to come by in the actual world. Large datasets for 3D reconstruction are expected to emerge in the future. One of the interesting future paths for study in depth estimation is emerging new self-adoption algorithms that can adapt to changing circumstances in real-time or with minimal supervision.

IX. SUMMARY

This analysis reveals significant heterogeneity in available datasets in terms of size (ranging from 5 to >1,800 classes), sensors used, image quality, and so on. Because of this variation, there is a dataset available for many research issues, but it is not always straightforward for researchers to choose the optimal alternative. This analysis not only serves to help researchers find the right dataset and loss function, but it also makes suggestions for establishing new ones. Because there are so many features that researchers can be interested in, presenting a global summary in the form of a research article is challenging. According to the bibliometric analysis, the KITTI dataset is the most cited, followed by CITYSCAPES and NYU-V2 datasets. As a

result, it is recommended that these datasets be used as benchmarks when comparing approaches to the published SoA. Furthermore, a license signed by a researcher is sufficient to get these datasets, as opposed to the signature of the institutional legal representative, which is normally requested by others. It's best to use datasets developed for specific challenges or competitions for comparative research because they come with a standardized evaluation methodology. MOBILE-RGBD is a tool for evaluating depth images obtained by smartphone cameras. FACESCAPE is a framework for studying 3D reconstruction and detection. There are 360⁰ and WEB STEREO VIDEO to examine combinations of multiple modalities. Reference [68] has put a lot of effort into developing publicly available datasets, in addition to KITTI and CITYSCAPES. Their website contains 102 high-quality datasets (plus more from other modalities), making it the most comprehensive web resource the authors found. Although the bibliometric analysis showed that these datasets are not as popular as those at KITTI or CITYSCAPES, NYU-V2 and did not cover the depth estimation-based research, it is encouraged that the academics explore them further.

X. RECOMMENDATION FOR BUILDING A COMPREHENSIVE DATASETS

Various scientific groups have explored important aspects of gathering and distributing research data.

- Plan availability for years to come - In the field of depth estimation, the acceptance of a new benchmark is typically difficult. It is critical to allocate resources for database distribution for several years into the future in order to maintain the database's availability. The most important resources are (i) technical – a solid URL for the promoting website as well as the infrastructure to keep it available – and (ii) personal – a designated person responsible for licensing maintenance as well as answering any problems that prospective users may encounter.
- Make access simple - We discovered that databases that include licenses that can be signed by individual academics are more popular. For young researchers, requiring the signature of the legal institutional representative, especially in a college environment (usually the rector), is a substantial barrier. Instead, they frequently choose to develop their own database. If an institutional representative's signature is required, we recommend posting the whole license agreement as well as a sample of the database images on the project website. This aids in determining whether the database is appropriate for a certain research project before beginning the administrative procedures required to secure the requisite approvals.
- Include a statistically relevant number of samples Acquiring and handling test subjects is one of the most challenging tasks when creating a biometric database. The number of subjects included should be as large as possible; however, there is always a minimum size

for obtaining statistically relevant results. Although this minimum is difficult to quantify for the general case, the statistical significance of 100 samples obtained from the same subjects is not the same as 1000 samples obtained from 100 different subjects.

- Make the database unique - Many authors who use a database in one publication continue to use it in subsequent publications. A database is often used to investigate particular qualities or problems in a methodical manner, as we have seen in earlier sections. A successful database should assist users in coming up with new research findings and conclusions. As a result, the database should be able to meet the needs of new study areas where benchmarks have yet to be created. With this review, the authors hope to aid in this work by making the demands more apparent to database designers.
- Extensive protocol and setup description - Despite the fact that the majority of the datasets available were developed to test a specific hypothesis or for a certain study aim, researchers frequently suggest that the dataset can be beneficial for more than one research topic. It is critical to offer a detailed description of the technique and setup in order to maximize the dataset's potential. Important information, such as the wavelength of the setup lighting, the distance at which the images were captured, and descriptions of the sensor or optical system employed, is usually lacking, restricting the usability of the datasets.
- More Challenging Datasets - For depth estimation and instance segmentation, several large-scale image datasets have been generated. However, new complex datasets, as well as datasets for diverse types of images, are still needed. Datasets containing a large number of objects and overlapping objects would be quite useful for still images. This may make it possible to train models that are better at dealing with dense object scenarios and high overlaps between objects, which are typical in real life. With the growing popularity of 3D image depth reconstruction, particularly in autonomous vehicles and robotics, large-scale 3D image datasets are in high demand. The creation of these datasets is more difficult than that of their lower-dimensional equivalents. Existing datasets for 3D image depth estimation are often insufficiently large, and some are synthetic, therefore larger and more difficult 3D image datasets can be extremely beneficial.

XI. CONCLUSION

This paper provides a detail review of the depth datasets and loss functions developed in the field of computer vision for depth estimation problems. The publicly available depth datasets and depth-based loss functions have achieved impressive performance in various depth maps tasks based on deep learning networks. People detection and action recognition, faces and poses, perception-based navigation (i.e., street signs, roads), object and scene recognition, and medical

applications are among the five general categories in which the depth datasets are categorized. Each depth dataset's main properties and characteristics are described and compared. To generalize model results across different environments, a mixing approach for depth datasets is presented. In addition, depth estimation loss functions are briefly presented, which will facilitate in the training of deep learning depth estimation models on a variety of datasets for both short- and long-range depth map estimation. Three of the most popular datasets are evaluated using SoA deep learning-based depth estimation algorithms. Finally, there is a discussion of challenges and future research, as well as recommendations for creating comprehensive depth datasets, which will help researchers in choosing relevant datasets and loss functions for evaluating their results and methods.

The main aim of this survey paper is that, to speed up the research in depth estimation tasks and compare the results to SoA methodologies for use case applications, researchers in this discipline must first understand the appropriate depth datasets and loss functions. To improve generalization, researchers should incorporate various datasets during training, validation, and testing. However, when combining datasets with different features, caution is required. The network's design and building blocks are important, but its performance is mostly influenced by how it is trained, which requires a diverse dataset and an appropriate loss function.

REFERENCES

- [1] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proc. CVPR*, Jun. 2011, pp. 1521–1528.
- [2] C. Wang, S. Lucey, F. Perazzi, and O. Wang, "Web stereo video supervision for depth prediction from dynamic scenes," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 348–357.
- [3] W. Chen, Z. Fu, D. Yang, and J. Deng, "Single-image depth perception in the wild," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 730–738.
- [4] T. Schops, J. L. Schonberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3260–3269.
- [5] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, 2017.
- [6] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: Structured-light versus time-of-flight kinect," *Comput. Vis. Image Understand.*, vol. 139, pp. 1–20, Oct. 2015.
- [7] F. Khan, S. Hussain, S. Basak, J. Lemley, and P. Corcoran, "An efficient encoder–decoder model for portrait depth estimation from single images trained on pixel-accurate synthetic data," *Neural Netw.*, vol. 142, pp. 479–491, Oct. 2021.
- [8] X.-F. Han, H. Laga, and M. Bennamoun, "Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1578–1604, May 2021.
- [9] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Feb. 17, 2021, doi: [10.1109/TPAMI.2021.3059968](https://doi.org/10.1109/TPAMI.2021.3059968).
- [10] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 27, 2020, doi: [10.1109/TPAMI.2020.3019967](https://doi.org/10.1109/TPAMI.2020.3019967).
- [11] F. Khan, S. Salahuddin, and H. Javidnia, "Deep learning-based monocular depth estimation methods—A state-of-the-art review," *Sensors*, vol. 20, no. 8, p. 2272, Apr. 2020.
- [12] A. Bhoi, "Monocular depth estimation: A survey," 2019, *arXiv:1901.09402*.
- [13] C. Zhao, Q. Sun, C. Zhang, Y. Tang, and F. Qian, "Monocular depth estimation based on deep learning: An overview," *Sci. China Technol. Sci.*, vol. 63, pp. 1612–1627, Jun. 2020.
- [14] Y. Ming, X. Meng, C. Fan, and H. Yu, "Deep learning for monocular depth estimation: A review," *Neurocomputing*, vol. 438, pp. 14–33, May 2021.
- [15] A. Mertan, D. J. Duff, and G. Unal, "Single image depth estimation: An overview," 2021, *arXiv:2104.06456*.
- [16] H. Song, J. Hong, H. Choi, and J. Min, "Concrete delamination depth estimation using a noncontact MEMS ultrasonic sensor array and an optimization approach," *Appl. Sci.*, vol. 11, no. 2, p. 592, Jan. 2021.
- [17] J. K. Devine, E. D. Chinoy, R. R. Markwald, L. P. Schwartz, and S. R. Hursh, "Validation of Zulu watch against polysomnography and actigraphy for on-wrist sleep-wake determination and sleep-depth estimation," *Sensors*, vol. 21, no. 1, p. 76, Dec. 2020.
- [18] P. Liu, Z. Zhang, Z. Meng, and N. Gao, "Monocular depth estimation with joint attention feature distillation and wavelet-based loss function," *Sensors*, vol. 21, no. 1, p. 54, Dec. 2020.
- [19] P. N. V. R. Koutilya, H. Zhou, and D. Jacobs, "SharinGAN: Combining synthetic and real data for unsupervised geometry estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 13974–13983.
- [20] J. Spencer, R. Bowden, and S. Hadfield, "DeFeat-Net: General monocular depth via simultaneous unsupervised representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 14402–14413.
- [21] P. Corcoran and H. Javidnia, "Accurate depth map estimation from small motions," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Oct. 2017, pp. 2453–2461.
- [22] S. Bazrafkan, H. Javidnia, and J. Lemley, "Semiparallel deep neural network hybrid architecture: First application on depth from monocular camera," *J. Electron. Imag.*, vol. 27, no. 4, p. 1, Aug. 2018, doi: [10.1117/1.jei.27.4.043041](https://doi.org/10.1117/1.jei.27.4.043041).
- [23] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "On the uncertainty of self-supervised monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 3227–3237.
- [24] K. Xian, J. Zhang, O. Wang, L. Mai, Z. Lin, and Z. Cao, "Structure-guided ranking loss for single image depth prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 611–620.
- [25] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3D packing for self-supervised monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 2485–2494.
- [26] H. Javidnia and P. Corcoran, "Real-time automotive street-scene mapping through fusion of improved stereo depth and fast feature detection algorithms," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, 2017, pp. 225–228.
- [27] L. Spinello and K. O. Arras, "People detection in RGB-D data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 3838–3843.
- [28] E. Cippitelli, E. Gambi, S. Gasparrini, and S. Spinsante, "TST fall detection dataset v2," IEEE Dataport, Tech. Rep., 2016, doi: [10.21227/H2VC7J](https://doi.org/10.21227/H2VC7J).
- [29] Z. Li, T. Dekel, F. Cole, R. Tucker, N. Snively, C. Liu, and W. T. Freeman, "Learning the depths of moving people by watching frozen people," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4521–4530.
- [30] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley MHAD: A comprehensive multimodal human action database," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2013, pp. 53–60.
- [31] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Comput. Methods Programs Biomed.*, vol. 117, no. 3, pp. 489–501, 2014.
- [32] D. Vaufraydaz and A. Nègre, "MobileRGBD, an open benchmark corpus for mobile RGB-D related algorithms," in *Proc. 13th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, Dec. 2014, pp. 1668–1673.
- [33] G. Fanelli, T. Weise, J. Gall, and L. Van Gool, "Real time head pose estimation from consumer depth cameras," in *Proc. Joint Pattern Recognit. Symp.*, 2011, pp. 101–110.
- [34] R. Min, N. Kose, and J.-L. Dugelay, "KinectFaceDB: A kinect database for face recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 11, pp. 1534–1548, Nov. 2014, doi: [10.1109/TSMC.2014.2331215](https://doi.org/10.1109/TSMC.2014.2331215).
- [35] G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara, "POSEidon: Face-from-depth for driver pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5494–5503, doi: [10.1109/CVPR.2017.583](https://doi.org/10.1109/CVPR.2017.583).

- [36] H. Yang, H. Zhu, Y. Wang, M. Huang, Q. Shen, R. Yang, and X. Cao, "FaceScape: A large-scale high quality 3D face dataset and detailed rig-gable 3D face prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 601–610.
- [37] N. Erdogmus and S. Marcel, "Spoofing in 2D face recognition with 3D masks and anti-spoofing with kinect," in *Proc. IEEE 6th Int. Conf. Biometrics, Theory, Appl. Syst. (BTAS)*, Sep. 2013, pp. 1–8.
- [38] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361, doi: [10.1109/CVPR.2012.6248074](https://doi.org/10.1109/CVPR.2012.6248074).
- [39] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road scene segmentation from a single image," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 376–389.
- [40] R. Zhang, S. A. Candra, K. Vetter, and A. Zakhor, "Sensor fusion for semantic segmentation of urban scenes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1850–1857.
- [41] G. Ros, S. Ramos, M. Granados, A. Bakhtiyar, D. Vazquez, and A. M. Lopez, "Vision-based offline-online perception paradigm for autonomous driving," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 231–238.
- [42] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3213–3223.
- [43] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou, "DrivingStereo: A large-scale dataset for stereo matching in autonomous driving scenarios," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 899–908.
- [44] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNs," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 11–20.
- [45] Z. Bauer, F. Gomez-Donoso, E. Cruz, S. Orts-Escolano, and M. Cazorla, "UASOL, a large-scale high-resolution outdoor stereo dataset," *Sci. Data*, vol. 6, no. 1, pp. 1–14, Dec. 2019.
- [46] J. Hidalgo-Carrió, D. Gehrig, and D. Scaramuzza, "Learning monocular dense depth from events," in *Proc. Int. Conf. 3D Vis. (3DV)*, Nov. 2020, pp. 534–542.
- [47] A. Sharma and J. Ventura, "Unsupervised learning of depth and ego-motion from cylindrical panoramic video," in *Proc. IEEE Int. Conf. Artif. Intell. Virtual Reality (AIVR)*, Dec. 2019, pp. 558–587.
- [48] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 746–760.
- [49] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2D–3D-semantic data for indoor scene understanding," 2017, *arXiv:1702.01105*.
- [50] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using SfM and object labels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1625–1632.
- [51] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 567–576.
- [52] Z. Li and N. Snavely, "MegaDepth: Learning single-view depth prediction from internet photos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2041–2050, doi: [10.1109/CVPR.2018.00218](https://doi.org/10.1109/CVPR.2018.00218).
- [53] I. Vasiljevic, N. Kolkin, S. Zhang, R. Luo, H. Wang, F. Z. Dai, A. F. Daniele, M. Mostajabi, S. Basart, M. R. Walter, and G. Shakhnarovich, "DIODE: A dense indoor and outdoor DEpth dataset," 2019, *arXiv:1908.00463*.
- [54] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. German Conf. Pattern Recognit.*, 2014, pp. 31–42.
- [55] H.-A. Le, T. Mensink, P. Das, S. Karaoglu, and T. Gevers, "EDEN: Multimodal synthetic dataset of enclosed GARden scenes," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2021, pp. 1579–1589.
- [56] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1746–1754.
- [57] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3D: Learning from RGB-D data in indoor environments," 2017, *arXiv:1709.06158*.
- [58] A. R. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3712–3722.
- [59] Q.-H. Pham, M. A. Uy, B.-S. Hua, D. T. Nguyen, G. Roig, and S.-K. Yeung, "LCD: Learned cross-domain descriptors for 2D–3D matching," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11856–11864.
- [60] N. Zioulis, A. Karakottas, D. Zarpalas, and P. Daras, "OmniDepth: Dense depth estimation for indoors spherical panoramas," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 448–465.
- [61] D. Wang, "MineNav: An expandable synthetic dataset based on minecraft for aircraft visual navigation," 2020, *arXiv:2008.08454*.
- [62] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3D scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, May 2009.
- [63] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.
- [64] K. B. Ozyoruk, G. I. Gokceler, T. L. Bobrow, G. Coskun, K. Incetan, Y. Almalioglu, F. Mahmood, E. Curto, L. Perdigoto, M. Oliveira, H. Sahin, H. Araujo, H. Alexandrino, N. J. Durr, H. B. Gilbert, and M. Turan, "EndoSLAM dataset and an unsupervised monocular visual odometry and depth estimation approach for endoscopic videos," *Med. Image Anal.*, vol. 71, Jul. 2021, Art. no. 102058.
- [65] V. Srivastav, T. Issenhuth, A. Kadkhodamohammadi, M. de Mathelin, A. Gangi, and N. Padoy, "MFOR: A multi-view RGB-D operating room dataset for 2D and 3D human pose estimation," 2018, *arXiv:1808.08180*.
- [66] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy, "EndoNet: A deep architecture for recognition tasks on laparoscopic videos," *IEEE Trans. Med. Imag.*, vol. 36, no. 1, pp. 86–97, Jan. 2017.
- [67] N. L. Rodas, F. Barrera, and N. Padoy, "See it with your own eyes: Markerless mobile augmented reality for radiation awareness in the hybrid room," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 2, pp. 429–440, Feb. 2017.
- [68] B. Ummerhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "DeMoN: Depth and motion network for learning monocular stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5038–5047.
- [69] J. M. Facil, B. Ummerhofer, H. Zhou, L. Montesano, T. Brox, and J. Civera, "CAM-Convs: Camera-aware multi-scale convolutions for single-view depth," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 11826–11835.
- [70] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," 2018, *arXiv:1810.04650*.
- [71] S. Niklaus, L. Mai, J. Yang, and F. Liu, "3D ken burns effect from a single image," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–15, Nov. 2019.
- [72] S. F. Bhat, I. Alhashim, and P. Wonka, "AdaBins: Depth estimation using adaptive bins," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 4009–4018.
- [73] G. Yang, H. Tang, M. Ding, N. Sebe, and E. Ricci, "Transformers solve the limited receptive field for monocular depth prediction," 2021, *arXiv:2103.12091*.
- [74] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," 2019, *arXiv:1907.10326*.
- [75] J. Chang and G. Wetzstein, "Deep optics for monocular depth estimation and 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 10193–10202.
- [76] M. Song, S. Lim, and W. Kim, "Monocular depth estimation using Laplacian pyramid-based depth residuals," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 11, pp. 4381–4393, Nov. 2021.
- [77] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 12179–12188.
- [78] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011, doi: [10.1109/CVPR.2018.00214](https://doi.org/10.1109/CVPR.2018.00214).
- [79] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," 2020, *arXiv:2005.10821*.
- [80] Y. Yuan, X. Chen, X. Chen, and J. Wang, "Segmentation transformer: Object-contextual representations for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 1, 2021.
- [81] R. Mohan and A. Valada, "EfficientPS: Efficient panoptic segmentation," *Int. J. Comput. Vis.*, vol. 129, no. 5, pp. 1551–1579, May 2021.

- [82] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 12475–12485.
- [83] X. Zhang, H. Xu, H. Mo, J. Tan, C. Yang, L. Wang, and W. Ren, "DCNAS: Densely connected neural architecture search for semantic image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 13956–13967.
- [84] J.-H. Lee and C.-S. Kim, "Multi-loss rebalancing algorithm for monocular depth estimation," in *Proc. 16th Eur. Conf.*, Glasgow, U.K., Aug. 2020, pp. 785–801.
- [85] J. Y. Jason, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 3–10.



FAISAL KHAN received the B.S. degree in mathematics from the University of Malakand, Chankdara, Pakistan, in 2015, and the M.Phil. degree in mathematics from Hazara University Mansehra, Pakistan, in 2017. He is currently pursuing the Ph.D. degree with the National University of Ireland Galway (NUIG). He is with FotoNation/Xperi. His research interests include machine learning using deep neural networks for tasks related to computer vision, including depth estimation and 3-D reconstruction.



SHAHID HUSSAIN received the B.S. degree in mathematics and the M.Sc. degree in computer science from the University of Peshawar, Pakistan, in 2002 and 2005, respectively, and the M.S. and Ph.D. degrees in computer engineering from Jeonbuk National University, South Korea, in 2016 and 2020, respectively. He had worked as a Postdoctoral Researcher at the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea. He is currently working as a Postdoctoral Research Fellow at the National University of Ireland, Galway, Ireland. His research interests include smart grid, energy management, electric vehicles, optimization algorithms, micro-grid operations, distributed energy resources, peer-to-peer energy trading, and image processing using fuzzy logic, game theory, ontologies, AI, and block-chain approaches and technologies. He was awarded with the Jeonbuk National University Presidential Award for academic excellence during his Ph.D. studies.



SHUBHAJIT BASAK (Graduate Student Member, IEEE) received the B.Tech. degree in electronics and communication engineering from the West Bengal University of Technology, India, in 2011, and the M.Sc. degree in computer science from the National University of Ireland Galway, Ireland, in 2018, where he is currently pursuing the Ph.D. degree in computer science. He had more than six years of industrial experience as a Software Development Professionalist. He is with FotoNation/Xperi. His research interest includes deep learning tasks related to computer vision.



MOHAMED MOUSTAFA (Member, IEEE) received the B.Sc. degree (Hons.) in computer science and information technology from the National University of Ireland, Galway, in 2021, where he is currently pursuing the Ph.D. degree in electrical and electronics engineering, as part of his employment-based postgraduate programme jointly funded by the Irish Research Council and Xperi Corporation. He is employed at Xperi Corporation. His research interests include computer vision, deep learning, embedded systems, edge-AI, and their applications for health monitoring. During his undergraduate studies, he was awarded the University Scholar Title three years in a row by the university.



PETER CORCORAN (Fellow, IEEE) currently the Personal Chair in electronic engineering at the College of Science and Engineering, National University of Ireland Galway. He was a Co-Founder in several start-up companies, notably FotoNation, now the Imaging Division of Xperi Corporation. He has over 600 technical publications and patents, over 100 peer-reviewed journal articles, 120 international conference papers, and a co-inventor of more than 300 granted U.S. patents. He is an IEEE Fellow recognized for his contributions to digital camera technologies, notably in-camera red-eye correction, and facial detection. He is a member of the IEEE Consumer Electronics Society for over 25 years. He is the Editor-in-Chief and the Founding Editor of *IEEE Consumer Electronics Magazine*.

...

Chapter 7

Conclusion and Future Work

In this chapter, we summarize the main contributions of this dissertation along with their limitations and future work with respect to each of these contributions.

7.1 Contribution to the Generation of synthetic Face Data

In Chapter 2, a synthetic facial data generation pipeline is proposed based on low-cost asset creation software and an open-source CG tool. With the proposed pipeline, it is possible not only to create a large number of samples, but also to add a large number of variations and randomness by adding virtual scene augmentation. With the help of the proposed methodology, we have then generated a large-face dataset with 100 identities and five different facial expressions (Appendix A). The whole dataset has more than 300k sample RGB face images with their head pose and raw face depth data as their ground truth annotations. This kind of large synthetic face dataset can be useful to train deep learning models for head pose estimation and monocular depth estimation tasks. We have also released the virtual human models and the data generation pipeline code written in Python. With the help of this open-sourced code base and the synthetic models, one can generate a large amount of full-body synthetic data with sufficient augmentations.

In future work, as we have access to the full body models, the dataset can be extended to collect different ground truth annotations like face segmentation, facial landmarks, full body activity recognition, etc. Though the proposed pipeline can create a large amount of face dataset, the rendered face images still look different from the real face images. So another future direction can be to extend the data with more realistic face textures generated by generative methods like StyleGAN or diffusion, which will reduce the domain gap between the real and synthetic faces.

7.2 Contribution to the Validation of the generated synthetic Face Data

In Chapter 3, we validate the synthetic data that was generated in the previous task. As we have collected two different ground truth annotations (i.e., head pose and face depth), we have chosen the facial analysis tasks associated with these two ground truths.

We first train a SOTA HPE model with only the synthetic head pose data generated by our method and validate its performance against a real dataset. The model trained only on our synthetic data gives a competitive result when compared with other SOTA HPE methods over some ranges of head pose. Though it gives a good result in some narrow head pose angles, it performs poorly for profile face images. So, to improve its performance, we then apply the transfer learning approach, a common training paradigm for training models on synthetic data. We first train the model on our synthetic data and then fine-tune it with a small set of real data. The fine-tuned model is able to surpass the result of the previous SOTA method that follows a similar transfer learning approach by a large margin while using only 10 percent of the real data compared to the previous SOTA method.

Though applying transfer learning to our data gives SOTA performance, we then focus on training the model without labeled real data. The major reason behind the poor performance of the model, when trained only on synthetic data, is the domain gap between the real and synthetic face images. So to reduce the domain gap, we then introduce an adversarial learning method where we train the model simultaneously on synthetic data against the main objective function of learning the head pose, and on unlabelled real data on an adversarial objective to reduce the gap between the real to the synthetic domain. This synthetic to real domain adaptation technique normally produces good results for classification tasks, as these have discrete label spaces, and matching source label clusters to target label clusters are comparatively easier. But for the HPE tasks where the label spaces are continuous, the traditional adversarial domain adaptation does not give good results. So we introduce a sampling methodology to sample the label spaces from the target real domain so that it keeps close to source synthetic labels (Appendix B). With the proposed adversarial domain adaptation learning, we are able to achieve near SOTA results in HPE tasks and show the potential of this technique for learning regression tasks from solely synthetic label spaces. We also validate the face depth data against the monocular depth estimation task. We propose an efficient encoder-decoder-based model with a hybrid loss function for accurate monocular facial depth estimation. The model is competitive with the other SOTA methods but significantly smaller in size and computational complexity, which makes it suitable for deployment in edge-AI applications.

While working with the real head pose, we have found visually that some of the frames with high yaw, pitch, and roll angles have errors in their ground truth annotations which are difficult to measure. As the collected synthetic ground truth head pose has accurate annotations, as a future work, a cross-validation approach can be applied to the real and synthetic datasets to identify the errors in the real ground truth annotations. Additionally, we can extend the adversarial domain adaptation approach to the facial depth estimation task as well to improve the depth estimation model performance earning from synthetic data only.

7.3 Contribution to the unsupervised Face Reconstruction from a single Image

In Chapter 4, we extend the 3D facial analysis task by proposing a weakly supervised approach to learning the 3D face structure from a single 2D face image. In the previous work of learning the facial depth from synthetic data, we have found that though we are able to learn the 3D structure of the face, estimating an accurate 3D face from the monocular face images is still not possible by learning from synthetic data only. So we introduce a hierarchical feature fusion-based vision transformer backbone for 3D facial feature extraction and propose unsupervised learning of the 3D face structure by introducing a differential renderer in the training pipeline. We train our network without any ground truth 3D face scan data and only utilize a large face dataset with its corresponding 3DMM parameters. We also introduce a hybrid loss function that combines both supervised and unsupervised objective functions. As per our knowledge, we are the first to introduce a vision transformer backbone to the face reconstruction task. Through both qualitative and quantitative results, we have shown that our method has achieved competitive results with other SOTA methods.

We have used the vision transformer as the feature extractor, which is by nature memory intensive and larger in size. This hinders its use and deployment in edge devices. In future work, we can use different width and depth pruning mechanisms to remove the unimportant units in the network and reduce the model size and computational complexity. Also, for supervised training, we have used the 3DMM parameters from the 300W-LP dataset, which mostly encodes the global facial deformations. So it fails to recover low-dimensional details like wrinkles. To recover these low-level features, we can use graph convolutions which are very effective in modeling the neighborhood vertex information. On the other hand, transformers are very good at predicting non-local interactions. So we can infuse the graph convolution into the transformer architecture to recover the fine-grain details of the face.

7.4 Contribution to the Data Generation and Model Building for dense Face Landmark Estimation

In Chapter 5, we extend the previous work to create a lightweight face shape predictor. Instead of predicting a dense shape mesh, we focus on predicting dense face key points. We assume that from a dense facial landmark it is possible to predict the full face shape information. Currently, there is no face dataset publicly available that has dense landmarks available as ground truth. So we first introduce a pipeline to create a dense landmark of 520 key points sampled from a UV position map data that can be extracted by fitting a face model to a face with the help of its 3DMM parameters. Using the newly created ground truth dense landmark data, we train a lightweight model with the MobilenetV2 backbone that predicts the dense landmarks from a single-face image. As there is no real data available with dense landmark ground truth for evaluation, we evaluate our model against the 3D face alignment task with 68 3D face key points. From the result, we find the trained model performs well when compared to other SOTA tasks when evaluated on the 3D face alignment task. As we use the lightweight Mobilenet backbone as the feature extractor, the overall model size and the memory requirement (FLOPs) are comparatively much smaller than the other SOTA backbones.

As future work, we can extend this to utilize more lightweight backbones like VarGFaceNet [156] to optimize the model further. We can also apply different knowledge distillation methods to pass task-specific features from a large, high-performing network. As we are trying to learn the dense landmark, we can create a weighted graph network and apply graph learning to replace the normal convolutions with graph convolution to learn the relationship and dependencies of the neighbor landmarks. As we don't have access to any ground truth-dense landmark dataset, we evaluate our model against the 68 key points. To evaluate the whole face shape, we can fit an existing face base model with the help of the dense landmarks and evaluate the learned shape against the ground truth face scans available publicly.

References

- [1] 3d character maker | character creator. <https://www.reallusion.com/character-creator/>. (Accessed on 03/18/2023).
- [2] blender.org - home of the blender project - free and open 3d creation software. <https://www.blender.org/>. (Accessed on 03/18/2023).
- [3] Character animation software - one stop solution | iclone. <https://www.reallusion.com/iclone/>. (Accessed on 03/18/2023).
- [4] Abdelmounaime, S. and Dong-Chen, H. (2013). New brodatz-based image databases for grayscale color and multiband texture analysis. *International Scholarly Research Notices*, 2013.
- [5] Allen, L., O’Connell, A., and Kiermer, V. (2019). How can we ensure visibility and diversity in research contributions? how the contributor role taxonomy (credit) is helping the shift from authorship to contributorship. *Learned Publishing*, 32(1):71–74.
- [6] Alp Guler, R., Trigeorgis, G., Antonakos, E., Snape, P., Zafeiriou, S., and Kokkinos, I. (2017). Densereg: Fully convolutional dense shape regression in-the-wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6799–6808.
- [7] Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*.
- [8] Bagdanov, A. D., Del Bimbo, A., and Masi, I. (2011). The florence 2d/3d hybrid face dataset. In *Proceedings of the 2011 joint ACM workshop on Human gesture and behavior understanding*, pages 79–80.
- [9] Bak, S., Carr, P., and Lalonde, J.-F. (2018). Domain adaptation through synthesis for unsupervised person re-identification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 189–205.
- [10] Bao, L., Lin, X., Chen, Y., Zhang, H., Wang, S., Zhe, X., Kang, D., Huang, H., Jiang, X., Wang, J., et al. (2021). High-fidelity 3d digital human head creation from rgb-d selfies. *ACM Transactions on Graphics (TOG)*, 41(1):1–21.
- [11] Bas, A., Huber, P., Smith, W. A., Awais, M., and Kittler, J. (2017). 3d morphable models as spatial transformer networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 904–912.

- [12] Basak, S., Corcoran, P., Khan, F., McDonnell, R., and Schukat, M. (2021a). Learning 3d head pose from synthetic data: A semi-supervised approach. *IEEE Access*, 9:37557–37573.
- [13] Basak, S., Corcoran, P., McDonnell, R., and Schukat, M. (2022). 3d face-model reconstruction from a single image: A feature aggregation approach using hierarchical transformer with weak supervision. *Neural Networks*, 156:108–122.
- [14] Basak, S., Javidnia, H., Khan, F., McDonnell, R., and Schukat, M. (2020). Methodology for building synthetic datasets with virtual humans. In *2020 31st Irish Signals and Systems Conference (ISSC)*, pages 1–6.
- [15] Basak, S., Khan, F., Javidnia, H., Corcoran, P., McDonnell, R., and Schukat, M. (2023). C3i-synface: A synthetic head pose and facial depth dataset using seed virtual human models. *Data in Brief*, page 109087.
- [16] Basak, S., Khan, F., McDonnell, R., and Schukat, M. (2021b). Learning accurate head pose for consumer technology from 3d synthetic data. In *2021 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6.
- [17] Beeler, T., Bickel, B., Beardsley, P., Sumner, B., and Gross, M. (2010). High-quality single-shot capture of facial geometry. In *ACM SIGGRAPH 2010 papers*, pages 1–9.
- [18] Bhagavatula, C., Zhu, C., Luu, K., and Savvides, M. (2017). Faster than real-time facial alignment: A 3d spatial transformer network approach in unconstrained poses. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3980–3989.
- [19] Bigioi, D., Basak, S., Stypułkowski, M., Zieba, M., Jordan, H., McDonnell, R., and Corcoran, P. (2024). Speech driven video editing via an audio-conditioned diffusion model. *Image and Vision Computing*, page 104911.
- [20] Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999*, pages 187–194. Association for Computing Machinery, Inc.
- [21] Borghi, G., Venturelli, M., Vezzani, R., and Cucchiara, R. (2017). Poseidon: Face-from-depth for driver pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4661–4670.
- [22] Bozkir, E., Ünal, A. B., Akgün, M., Kasneci, E., and Pfeifer, N. (2020). Privacy preserving gaze estimation using synthetic images via a randomized encoding based framework. In *ACM symposium on eye tracking research and applications*, pages 1–5.
- [23] Bregler, C., Covell, M., and Slaney, M. (1997). Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 353–360.
- [24] Bulat, A. and Tzimiropoulos, G. (2016a). Convolutional aggregation of local evidence for large pose face alignment.

- [25] Bulat, A. and Tzimiropoulos, G. (2016b). Two-stage convolutional part heatmap regression for the 1st 3d face alignment in the wild (3dfaw) challenge. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14*, pages 616–624. Springer.
- [26] Bulat, A. and Tzimiropoulos, G. (2017). How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *Proceedings of the IEEE international conference on computer vision*, pages 1021–1030.
- [27] Cao, C., Chai, M., Woodford, O., and Luo, L. (2018). Stabilized real-time face tracking via a learned dynamic rigidity prior. *ACM Transactions on Graphics (TOG)*, 37(6):1–11.
- [28] Cao, X., Wei, Y., Wen, F., and Sun, J. (2014). Face alignment by explicit shape regression. *International journal of computer vision*, 107:177–190.
- [29] Chen, C.-F. R., Fan, Q., and Panda, R. (2021). Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366.
- [30] Chen, L., Li, Z., Maddox, R. K., Duan, Z., and Xu, C. (2018). Lip movements generation at a glance. In *Proceedings of the European conference on computer vision (ECCV)*, pages 520–535.
- [31] Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. (2020). Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*.
- [32] Chen, S., Liu, Z., Liu, J., and Wang, L. (2022). Talking head generation driven by speech-related facial action units and audio-based on multimodal representation fusion. *arXiv preprint arXiv:2204.12756*.
- [33] Chowdary, M. K., Nguyen, T. N., and Hemanth, D. J. (2021). Deep learning-based facial emotion recognition for human–computer interaction applications. *Neural Computing and Applications*, pages 1–18.
- [34] Chung, J. S., Jamaludin, A., and Zisserman, A. (2017). You said that? *arXiv preprint arXiv:1705.02966*.
- [35] Cudeiro, D., Bolkart, T., Laidlaw, C., Ranjan, A., and Black, M. J. (2019). Capture, learning, and synthesis of 3d speaking styles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10101–10111.
- [36] Dai, D., Sakaridis, C., Hecker, S., and Van Gool, L. (2020). Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding. *International Journal of Computer Vision*, 128:1182–1204.
- [37] Daněček, R., Black, M. J., and Bolkart, T. (2022). Emoca: Emotion driven monocular face capture and animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20311–20322.
- [38] Deng, J., Guo, J., Ververas, E., Kotsia, I., and Zafeiriou, S. (2020). Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5203–5212.

- [39] Deng, J., Guo, J., Xue, N., and Zafeiriou, S. (2019a). Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699.
- [40] Deng, Y., Yang, J., Xu, S., Chen, D., Jia, Y., and Tong, X. (2019b). Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0.
- [41] Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794.
- [42] Dhere, S., Rathod, S. B., Aarankalle, S., Lad, Y., and Gandhi, M. (2020). A review on face reenactment techniques. In *2020 International Conference on Industry 4.0 Technology (I4Tech)*, pages 191–194. IEEE.
- [43] Ding, H., Liu, C., Wang, S., and Jiang, X. (2021). Vision-language transformer and query generation for referring segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16321–16330.
- [44] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- [45] Dutta, T. (2012). Evaluation of the kinect™ sensor for 3-d kinematic measurement in the workplace. *Applied ergonomics*, 43(4):645–649.
- [46] Fan, Y., Lin, Z., Saito, J., Wang, W., and Komura, T. (2022). Faceformer: Speech-driven 3d facial animation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18770–18780.
- [47] Fanelli, G., Dantone, M., Gall, J., Fossati, A., and Van Gool, L. (2013). Random forests for real time 3d face analysis. *International journal of computer vision*, 101:437–458.
- [48] Fang, Y., Liao, B., Wang, X., Fang, J., Qi, J., Wu, R., Niu, J., and Liu, W. (2021). You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34:26183–26197.
- [49] Feng, Y., Wu, F., Shao, X., Wang, Y., and Zhou, X. (2018a). Joint 3d face reconstruction and dense alignment with position map regression network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 534–551.
- [50] Feng, Z.-H., Kittler, J., Awais, M., Huber, P., and Wu, X.-J. (2018b). Wing loss for robust facial landmark localisation with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2235–2245.
- [51] Floater, M. S. (1997). Parametrization and smooth approximation of surface triangulations. *Computer aided geometric design*, 14(3):231–250.
- [52] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.

- [53] Gao, Q., Shen, X., and Niu, W. (2020a). Large-scale synthetic urban dataset for aerial scene understanding. *IEEE Access*, 8:42131–42140.
- [54] Gao, Z., Zhang, J., Guo, Y., Ma, C., Zhai, G., and Yang, X. (2020b). Semi-supervised 3D face representation learning from unconstrained photo collections. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, volume 2020-June, pages 1426–1435.
- [55] Garrido, P., Valgaerts, L., Rehmsen, O., Thormahlen, T., Perez, P., and Theobalt, C. (2014). Automatic face reenactment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4217–4224.
- [56] Garrido, P., Valgaerts, L., Sarmadi, H., Steiner, I., Varanasi, K., Perez, P., and Theobalt, C. (2015). Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. In *Computer graphics forum*, volume 34, pages 193–204. Wiley Online Library.
- [57] Gecer, B., Ploumpis, S., Kotsia, I., and Zafeiriou, S. (2019). Ganfit: Generative adversarial network fitting for high fidelity 3D face reconstruction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 1155–1164.
- [58] Genova, K., Cole, F., Maschinot, A., Sarna, A., Vlasic, D., and Freeman, W. T. (2018). Unsupervised Training for 3D Morphable Model Regression. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 8377–8386.
- [59] Goel, K., Gu, A., Donahue, C., and Ré, C. (2022). It’s raw! audio generation with state-space models. In *International Conference on Machine Learning*, pages 7616–7633. PMLR.
- [60] Goel, N., Amayuelas, A., Deshpande, A., and Sharma, A. (2021). The importance of modeling data missingness in algorithmic fairness: A causal perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7564–7573.
- [61] Gourier, N., Hall, D., and Crowley, J. L. (2004). Estimating face orientation from robust detection of salient facial structures. In *FG Net workshop on visual observation of deictic gestures*, volume 6, page 7. Citeseer.
- [62] Gu, J., Kwon, H., Wang, D., Ye, W., Li, M., Chen, Y.-H., Lai, L., Chandra, V., and Pan, D. Z. (2022). Multi-scale high-resolution vision transformer for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12094–12103.
- [63] Gu, J., Yang, X., De Mello, S., and Kautz, J. (2017). Dynamic facial analysis: From bayesian filtering to recurrent neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1548–1557.
- [64] Guo, J., Zhu, X., Yang, Y., Yang, F., Lei, Z., and Li, S. Z. (2020). Towards fast, accurate and stable 3d dense face alignment. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX*, pages 152–168. Springer.

- [65] Guo, Y., Cai, J., Jiang, B., Zheng, J., and Others (2018). Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *IEEE transactions on pattern analysis and machine intelligence*, 41(6):1294–1307.
- [66] Harvey, W., Naderiparizi, S., Masrani, V., Weilbach, C., and Wood, F. (2022). Flexible diffusion modeling of long videos. *arXiv preprint arXiv:2205.11495*.
- [67] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [68] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.
- [69] Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. (2022). Video diffusion models. *arXiv preprint arXiv:2204.03458*.
- [70] Höpfe, T., Mehrjou, A., Bauer, S., Nielsen, D., and Dittadi, A. (2022). Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696*.
- [71] Hu, G., Peng, X., Yang, Y., Hospedales, T. M., and Verbeek, J. (2017a). Frankenstein: Learning deep face representations using small data. *IEEE Transactions on Image Processing*, 27(1):293–303.
- [72] Hu, P., Ning, H., Qiu, T., Song, H., Wang, Y., and Yao, X. (2017b). Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things. *IEEE Internet of Things Journal*, 4(5):1143–1155.
- [73] Hwang, H., Jang, C., Park, G., Cho, J., and Kim, I.-J. (2021). Eldersim: A synthetic data generation platform for human action recognition in eldercare applications. *IEEE Access*.
- [74] Im, J.-H., Jeon, S.-Y., and Lee, M.-K. (2020). Practical privacy-preserving face authentication for smartphones secure against malicious clients. *IEEE Transactions on Information Forensics and Security*, 15:2386–2401.
- [75] Jaipuria, N., Zhang, X., Bhasin, R., Arafa, M., Chakravarty, P., Shrivastava, S., Manglani, S., and Murali, V. N. (2020). Deflating dataset bias using synthetic data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 772–773.
- [76] Jalalifar, S. A., Hasani, H., and Aghajan, H. (2018). Speech-driven facial reenactment using conditional generative adversarial networks. *arXiv preprint arXiv:1803.07461*.
- [77] Jeni, L. A., Cohn, J. F., and Kanade, T. (2015). Dense 3d face alignment from 2d videos in real-time. In *2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, volume 1, pages 1–8. IEEE.
- [78] Joshi, I., Grimmer, M., Rathgeb, C., Busch, C., Bremond, F., and Dantcheva, A. (2022). Synthetic data in human analysis: A survey. *arXiv preprint arXiv:2208.09191*.

- [79] Josifovski, J., Kerzel, M., Pregizer, C., Posniak, L., and Wermter, S. (2018). Object detection and pose estimation based on convolutional neural networks trained with synthetic data. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 6269–6276. IEEE.
- [80] Karkkainen, K. and Joo, J. (2021). Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1548–1558.
- [81] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119.
- [82] Kartynnik, Y., Ablavatski, A., Grishchenko, I., and Grundmann, M. (2019). Real-time facial surface geometry from monocular video on mobile gpus. *arXiv preprint arXiv:1907.06724*.
- [83] Khabarлак, K. and Koriashkina, L. (2021). Fast facial landmark detection and applications: A survey. *arXiv preprint arXiv:2101.10808*.
- [84] Khan, F., Basak, S., and Corcoran, P. (2021a). Accurate 2d facial depth models derived from a 3d synthetic dataset. In *2021 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6.
- [85] Khan, F., Basak, S., Javidnia, H., Schukat, M., and Corcoran, P. (2020). High-accuracy facial depth models derived from 3d synthetic data. In *2020 31st Irish Signals and Systems Conference (ISSC)*, pages 1–5.
- [86] Khan, F., Hussain, S., Basak, S., Lemley, J., and Corcoran, P. (2021b). An efficient encoder–decoder model for portrait depth estimation from single images trained on pixel-accurate synthetic data. *Neural Networks*, 142:479–491.
- [87] Khan, F., Hussain, S., Basak, S., Moustafa, M., and Corcoran, P. (2021c). A review of benchmark datasets and training loss functions in neural depth estimation. *IEEE Access*, 9:148479–148503.
- [88] Kim, H., Garrido, P., Tewari, A., Xu, W., Thies, J., Niessner, M., Pérez, P., Richardt, C., Zollhöfer, M., and Theobalt, C. (2018). Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):1–14.
- [89] Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. (2020). Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*.
- [90] Kortylewski, A., Egger, B., Schneider, A., Gerig, T., Morel-Forster, A., and Vetter, T. (2018). Empirically analyzing the effect of dataset biases on deep face recognition systems. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2093–2102.
- [91] Kumar, A., Kaur, A., and Kumar, M. (2019). Face detection techniques: a review. *Artificial Intelligence Review*, 52:927–948.

- [92] Kumar, R., Sotelo, J., Kumar, K., de Brébisson, A., and Bengio, Y. (2017). Obamanet: Photo-realistic lip-sync from text. *arXiv preprint arXiv:1801.01442*.
- [93] Lee, D.-T. and Schachter, B. J. (1980). Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242.
- [94] Li, S. and Deng, W. (2020). Deep facial expression recognition: A survey. *IEEE transactions on affective computing*, 13(3):1195–1215.
- [95] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.
- [96] Mayer, N., Ilg, E., Fischer, P., Hazirbas, C., Cremers, D., Dosovitskiy, A., and Brox, T. (2018). What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision*, 126(9):942–960.
- [97] Meng, L., Li, H., Chen, B.-C., Lan, S., Wu, Z., Jiang, Y.-G., and Lim, S.-N. (2022). Advait: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12309–12318.
- [98] Min, R., Kose, N., and Dugelay, J.-L. (2014). Kinectfacedb: A kinect database for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(11):1534–1548.
- [99] Morales, A., Piella, G., and Sukno, F. M. (2021). Survey on 3D face reconstruction from uncalibrated images. *Computer Science Review*, 40:100400.
- [100] Movshovitz-Attias, Y., Kanade, T., and Sheikh, Y. (2016). How useful is photo-realistic rendering for visual learning? In *European conference on computer vision*, pages 202–217. Springer.
- [101] Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. (2021). Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.
- [102] Nikolenko, S. I. (2019). Synthetic data for deep learning. *arXiv preprint arXiv:1909.11512*.
- [103] Nikolenko, S. I. (2021). *Synthetic data for deep learning*, volume 174. Springer.
- [104] Nowruzi, F. E., Kapoor, P., Kolhatkar, D., Hassanat, F. A., Laganieri, R., and Rebut, J. (2019). How much real data do we actually need: Analyzing object detection performance using synthetic and real data. *arXiv preprint arXiv:1907.07061*.
- [105] Park, S. J., Hong, S., Kim, D., Hussain, I., and Seo, Y. (2019). Intelligent in-car health monitoring system for elderly drivers in connected car. In *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018) Volume VI: Transport Ergonomics and Human Factors (TEHF), Aerospace Human Factors and Ergonomics 20*, pages 40–44. Springer.

- [106] Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T. (2009). A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pages 296–301. Ieee.
- [107] Queiroz, R., Cohen, M., Moreira, J. L., Braun, A., Júnior, J. C. J., and Musse, S. R. (2010). Generating facial ground truth with synthetic faces. In *2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images*, pages 25–31. IEEE.
- [108] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.
- [109] Raji, I. D., Gebru, T., Mitchell, M., Buolamwini, J., Lee, J., and Denton, E. (2020). Saving face: Investigating the ethical concerns of facial recognition auditing. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 145–151.
- [110] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.
- [111] Rashid, M., Gu, X., and Jae Lee, Y. (2017). Interspecies knowledge transfer for facial keypoint detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6894–6903.
- [112] Richardson, E., Sela, M., and Kimmel, R. (2016). 3d face reconstruction by learning from synthetic data. In *2016 fourth international conference on 3D vision (3DV)*, pages 460–469. IEEE.
- [113] Richardson, E., Sela, M., Or-El, R., and Kimmel, R. (2017). Learning detailed face reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1259–1268.
- [114] Roberts, M., Ramapuram, J., Ranjan, A., Kumar, A., Bautista, M. A., Paczan, N., Webb, R., and Susskind, J. M. (2021). Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922.
- [115] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695.
- [116] Ruan, Z., Zou, C., Wu, L., Wu, G., and Wang, L. (2021). Sadrnet: Self-aligned dual face regression networks for robust 3d dense face alignment and reconstruction. *IEEE Transactions on Image Processing*, 30:5793–5806.
- [117] Ryan, C., O’Sullivan, B., Elrasad, A., Cahill, A., Lemley, J., Kielty, P., Posch, C., and Perot, E. (2021). Real-time face & eye tracking and blink detection using event cameras. *Neural Networks*, 141:87–97.

- [118] Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., and Norouzi, M. (2022). Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10.
- [119] Sajjad, M., Khan, S., Hussain, T., Muhammad, K., Sangaiah, A. K., Castiglione, A., Esposito, C., and Baik, S. W. (2019). Cnn-based anti-spoofing two-tier multi-factor authentication system. *Pattern Recognition Letters*, 126:123–131.
- [120] Savran, A., Alyüz, N., Dibeklioglu, H., Çeliktutan, O., Gökberk, B., Sankur, B., and Akarun, L. (2008). Bosphorus database for 3d face analysis. In *Biometrics and Identity Management: First European Workshop, BIOID 2008, Roskilde, Denmark, May 7-9, 2008. Revised Selected Papers 1*, pages 47–56. Springer.
- [121] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 1, pages 519–528. IEEE.
- [122] Sheng, H., Cai, S., Liu, Y., Deng, B., Huang, J., Hua, X.-S., and Zhao, M.-J. (2021). Improving 3d object detection with channel-wise transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2743–2752.
- [123] Shi, Y., Ying, X., and Yang, J. (2022). Deep unsupervised domain adaptation with time series sensor data: A survey. *Sensors*, 22(15).
- [124] Shi, Y., Zhang, Z., Huang, K., Ma, W., and Tu, S. (2020). Human-computer interaction based on face feature localization. *Journal of Visual Communication and Image Representation*, 70:102740.
- [125] Siddiqui, M. F., Siddique, W. A., Ahmedh, M., and Jumani, A. K. (2020). Face detection and recognition system for enhancing security measures using artificial intelligence system. *Indian Journal of Science and Technology*, 13(09):1057–1064.
- [126] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR.
- [127] Sugano, Y., Matsushita, Y., and Sato, Y. (2014). Learning-by-synthesis for appearance-based 3d gaze estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1821–1828.
- [128] Suwajanakorn, S., Seitz, S. M., and Kemelmacher-Shlizerman, I. (2017). Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (ToG)*, 36(4):1–13.
- [129] Tewari, A., Zollhöfer, M., Garrido, P., Bernard, F., Kim, H., Pérez, P., and Theobalt, C. (2018). Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2549–2559.

- [130] Tewari, A., Zollhofer, M., Kim, H., Garrido, P., Bernard, F., Perez, P., and Theobalt, C. (2017). Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1274–1283.
- [131] Thies, J., Zollhöfer, M., Nießner, M., Valgaerts, L., Stamminger, M., and Theobalt, C. (2015). Real-time expression transfer for facial reenactment. *ACM Trans. Graph.*, 34(6):183–1.
- [132] Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., and Nießner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395.
- [133] Tiwari, H., Kurmi, V. K., Venkatesh, K., and Chen, Y.-S. (2022). Occlusion resistant network for 3d face reconstruction. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 813–822.
- [134] Tran, L. and Liu, X. (2018). Nonlinear 3d face morphable model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7346–7355.
- [135] Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., and Birchfield, S. (2018). Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977.
- [136] Tsirikoglou, A., Kronander, J., Wrenninge, M., and Unger, J. (2017). Procedural modeling and physically based rendering for synthetic data generation in automotive applications. *arXiv preprint arXiv:1710.06270*.
- [137] Tu, X., Zhao, J., Xie, M., Jiang, Z., Balamurugan, A., Luo, Y., Zhao, Y., He, L., Ma, Z., and Feng, J. (2020). 3d face reconstruction from a single image assisted by 2d face images in the wild. *IEEE Transactions on Multimedia*, 23:1160–1172.
- [138] Tuan Tran, A., Hassner, T., Masi, I., and Medioni, G. (2017). Regressing robust and discriminative 3d morphable models with a very deep neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5163–5172.
- [139] Vougioukas, K., Petridis, S., and Pantic, M. (2018). End-to-end speech-driven facial animation with temporal gans. *arXiv preprint arXiv:1805.09313*.
- [140] Vougioukas, K., Petridis, S., and Pantic, M. (2020). Realistic speech-driven facial animation with gans. *International Journal of Computer Vision*, 128:1398–1413.
- [141] Wang, J., Mueller, F., Bernard, F., Sorli, S., Sotnychenko, O., Qian, N., Otaduy, M. A., Casas, D., and Theobalt, C. (2020a). Rgb2hands: real-time tracking of 3d hand interactions from monocular rgb video. *ACM Transactions on Graphics (ToG)*, 39(6):1–16.
- [142] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al. (2020b). Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364.

- [143] Wang, J., Wu, Z., Chen, J., Han, X., Shrivastava, A., Lim, S.-N., and Jiang, Y.-G. (2022). Objectformer for image manipulation detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2364–2373.
- [144] Wang, L., Chen, W., Yang, W., Bi, F., and Yu, F. R. (2020c). A state-of-the-art review on image synthesis with generative adversarial networks. *IEEE Access*, 8:63514–63537.
- [145] Wang, M. and Deng, W. (2021). Deep face recognition: A survey. *Neurocomputing*, 429:215–244.
- [146] Wang, X., Wang, K., and Lian, S. (2020d). A survey on face data augmentation for the training of deep neural networks. *Neural computing and applications*, 32(19):15503–15531.
- [147] Wang, Y., Huang, R., Song, S., Huang, Z., and Huang, G. (2021). Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Advances in Neural Information Processing Systems*, 34:11960–11973.
- [148] Wang, Y., Liang, W., Shen, J., Jia, Y., and Yu, L.-F. (2019). A deep coarse-to-fine network for head pose estimation from synthetic data. *Pattern Recognition*, 94:196–206.
- [149] Weise, T., Bouaziz, S., Li, H., and Pauly, M. (2011). Realtime performance-based facial animation. *ACM transactions on graphics (TOG)*, 30(4):1–10.
- [150] Wood, E., Baltrušaitis, T., Hewitt, C., Dziadzio, S., Cashman, T. J., and Shotton, J. (2021). Fake it till you make it: face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3681–3691.
- [151] Wood, E., Baltrušaitis, T., Hewitt, C., Johnson, M., Shen, J., Milosavljević, N., Wilde, D., Garbin, S., Sharp, T., Stojiljković, I., Cashman, T., and Valentin, J. (2022a). 3D Face Reconstruction with Dense Landmarks. pages 160–177.
- [152] Wood, E., Baltrušaitis, T., Hewitt, C., Johnson, M., Shen, J., Milosavljević, N., Wilde, D., Garbin, S., Sharp, T., Stojiljković, I., et al. (2022b). 3d face reconstruction with dense landmarks. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*, pages 160–177. Springer.
- [153] Wood, E., Baltrušaitis, T., Morency, L.-P., Robinson, P., and Bulling, A. (2016). Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 131–138.
- [154] Wu, B.-F., Wu, B.-J., Tsai, B.-R., and Hsu, C.-P. (2022). A facial-image-based blood pressure measurement system without calibration. *IEEE Transactions on Instrumentation and Measurement*, 71:1–13.
- [155] Xie, L. and Liu, Z.-Q. (2007). Realistic mouth-synching for speech-driven talking face using articulatory modelling. *IEEE Transactions on Multimedia*, 9(3):500–510.

- [156] Yan, M., Zhao, M., Xu, Z., Zhang, Q., Wang, G., and Su, Z. (2019). Vargfacenet: An efficient variable group convolutional neural network for lightweight face recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0.
- [157] Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Shao, Y., Zhang, W., Cui, B., and Yang, M.-H. (2022). Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*.
- [158] Yang, T.-Y., Chen, Y.-T., Lin, Y.-Y., and Chuang, Y.-Y. (2019). Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1087–1096.
- [159] Yin, F., Zhang, Y., Cun, X., Cao, M., Fan, Y., Wang, X., Bai, Q., Wu, B., Wang, J., and Yang, Y. (2022). Styleheat: One-shot high-resolution editable talking face generation via pre-trained stylegan. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 85–101. Springer.
- [160] Yu, Q., Xia, Y., Bai, Y., Lu, Y., Yuille, A. L., and Shen, W. (2021). Glance-and-gaze vision transformer. *Advances in Neural Information Processing Systems*, 34:12992–13003.
- [161] Yu, R., Saito, S., Li, H., Ceylan, D., and Li, H. (2017). Learning dense facial correspondences in unconstrained images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4723–4732.
- [162] Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., and Grundmann, M. (2020a). Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*.
- [163] Zhang, P., Dai, X., Yang, J., Xiao, B., Yuan, L., Zhang, L., and Gao, J. (2021). Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2998–3008.
- [164] Zhang, T., Deng, L., Zhang, L., and Dang, X. (2020b). Deep learning in face synthesis: A survey on deepfakes. In *2020 IEEE 3rd International Conference on Computer and Communication Engineering Technology (CCET)*, pages 67–70. IEEE.
- [165] Zhang, X., Yin, L., Cohn, J. F., Canavan, S., Reale, M., Horowitz, A., and Liu, P. (2013). A high-resolution spontaneous 3d dynamic facial expression database. In *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, pages 1–6. IEEE.
- [166] Zhang, X., Yin, L., Cohn, J. F., Canavan, S., Reale, M., Horowitz, A., Liu, P., and Girard, J. M. (2014). Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706.
- [167] Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H., et al. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890.

-
- [168] Zhu, X., Lei, Z., Liu, X., Shi, H., and Li, S. Z. (2016). Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 146–155.
- [169] Zhu, X., Liu, X., Lei, Z., and Li, S. Z. (2017). Face alignment in full pose range: A 3d total solution. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):78–92.
- [170] Zollhöfer, M., Thies, J., Garrido, P., Bradley, D., Beeler, T., Pérez, P., Stamminger, M., Nießner, M., and Theobalt, C. (2018). State of the art on monocular 3d face reconstruction, tracking, and applications. In *Computer graphics forum*, volume 37, pages 523–550. Wiley Online Library.