



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

Title	Contributions to neural network models and training datasets for facial depth
Author(s)	Khan, Faisal
Publication Date	2023-03-27
Publisher	NUI Galway
Item record	http://hdl.handle.net/10379/17709

Downloaded 2024-04-26T19:12:24Z

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



Contributions to Neural Network Models and Training Datasets for Facial Depth Estimation from a Single Image



Faisal Khan

College of Engineering and Informatics
National University of Ireland, Galway

This dissertation is submitted for the degree of
Doctor of Philosophy

Supervisor: Prof. Peter Corcoran

March 2023

“Better than a thousand days of diligent study is one day with a great teacher. ”

– Japanese Proverb

Table of contents

List of figures	x
List of tables	xi
Nomenclature	xii
1 Introduction to the Single-Image/Monocular Depth Challenge	1
1.1 Introduction	1
1.2 Facial Depth Estimation Challenge	5
1.3 Overview of Contributions in this Thesis	8
1.3.1 Contribution to general Review Work on the broad Field of single-Image Depth Maps and SoA NN Models	9
1.3.2 Contribution to Building 2D Datasets from 3D Models with Pixel-Accurate Depth GT	10
1.3.3 Contribution to improved NN Models for monocular Facial Depth Maps: Training new CNN Models and their Evaluation and, as a Sub-Contribution, a detailed Comparison of SoA Models	11
1.4 Other Contributions	12
1.5 List of Publications & Datasets	13
1.6 Contribution Taxonomy	14
2 Contribution to General 'Review' Work on the Broad Field of Single-Image Depth Maps and Neural Models	16
2.1 Research Objectives	17
2.2 Summary and Discussions of Contributions	18
2.2.1 Deep Learning-based Monocular Depth Estimation Methods—a State-of-the-art Review	18
2.2.2 A Review of Benchmark Datasets and Training Loss Functions in Neural Depth Estimation	20

2.2.3	Towards Monocular Neural Facial Depth Estimation: Past, Present, and Future	21
3	Contribution to Building 2D Datasets from 3D Models with Pixel-accurate Depth GT	24
3.1	Synthetic Data and Tools	24
3.1.1	The Challenges of ‘Real-World’ Data	25
3.2	Research Objectives	26
3.3	Summary and Discussions of Contributions	30
3.3.1	The Future of Synthetic Data	30
4	Contributions to Improving the Accuracy of Facial Depth Estimation	32
4.1	Deep Learning Model for Portrait Depth Estimation from Single Images trained on Pixel-Accurate Synthetic Data	32
4.1.1	Research Objectives	32
4.2	A Robust Light-Weight Fused-Feature Encoder-Decoder Model for Monocular Facial Depth Estimation from Single Images Trained on Synthetic Data .	36
4.2.1	Research Objectives	36
4.2.2	Summary of Contributions	38
4.2.3	Discussion of Contributions	39
5	Additional Contributions	40
6	Conclusions and Future Works	42
	References	45
	Appendix A Deep Learning-Based Monocular Depth Estimation Methods—A State-of-the-Art Review	53
	Appendix B A Review of Benchmark Datasets and Training Loss Functions in Neural Depth Estimation	70
	Appendix C Towards Monocular Neural Facial Depth Estimation: Past, Present, and Future	96
	Appendix D High-Accuracy Facial Depth Models derived from 3D Synthetic Data	120

Appendix E	Accurate 2D Facial Depth Models Derived from a 3D Synthetic Dataset	126
Appendix F	An efficient encoder–decoder model for portrait depth estimation from single images trained on pixel-accurate synthetic data	133
Appendix G	A Robust Light-Weight Fused-Feature Encoder-Decoder Model for Monocular Facial Depth Estimation from Single Images Trained on Synthetic Data	147
Appendix H	Learning 3D Head Pose From Synthetic Data: A Semi-Supervised Approach	160
Appendix I	Methodology for Building Synthetic Datasets with Virtual Humans	178
Appendix J	Learning Accurate Head Pose for Consumer Technology From 3D Synthetic Data	185

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.

Faisal Khan
March 2023

Acknowledgements

I want to thank and praise practically everyone who has supported me in any manner during my PhD journey.

First of all, I want to thank Peter Corcoran, who is both my supervisor and mentor. It was Professor Corcoran who first accepted and advised me to apply for a PhD scholarship here at NUI Galway and ultimately College of Science and Engineering awarded me a Ph.D. position. I will always be grateful for the numerous possibilities you have provided me and still helping me with everything. You were always encouraging and gave me the confidence to believe I could complete the Ph.D. adventure. I would be always thankful for your advice, supportiveness, insightful and timely feedback, and ideas to improve my draft articles.

I can never feel any difficulty to talk about anything I wanted, and wish we will continue this wonderful journey. Again, many thanks. You are incredible, not only as a supervisor but also as a person, and I have learnt a lot from you and will continue to learn from you.

I did like to express my gratitude to Professor Chris Dainty. It is unusual to have the opportunity to interact with a scholar of your caliber, but you always had time for everybody.

I would like to thank Joe Lemley, who despite being a very busy manager in Xperi, always had the time to answer my questions, help me advance my research especially my writing skills. Also I would like to thank you for all the time we had together in our house. He invited me to a day trip to Cork which brought a lot of joy to me and it was my first in Ireland. A special thank goes to Snail Joe wife for all the wonderful moments and all the lunches and dinners we did together.

A special thanks goes to my best friends, Muhammad Ali Farooq, Waseem Shariff, Shubjhait Basak, Saqib Salauddin, Adnan Alahi, Muhammad Yahya, Haroon Zafer for all the wonderful moments we spent together, helping me out with paper drafting and proofreading the work, all the lunches and dinners we shared, funny and sometimes useful conversations we had, and jokes we cracked on each other.

Hossein Javidnia, you were an amazing mentor, taking me under your wing and helping me especially in the beginning of my PhD. It was a gift collaborating with you.

Thank you Joe Desbonnet for your knowledge and support. Thank you Amr Elrasad, Cain Ryan, Aoife McDonagh, Alin and Diana for the friendship and for all the help, all the lunchroom discussions and all the experiences we have shared together.

I would especially like to thank all of the dedicated professors and teachers from my past who helped me out during my academic career and inspired me to pursue independent research.

My colleagues and friends at Xperi, NUIG, and the C3 Imaging lab will always have my gratitude which I am delighted to see that is growing.

Many thanks to all of my Pakistani and Irish friends Saddam, Rehan, Abrar, Arsalan, Amjad, Frank, Una who have helped me over the years. Friends are the family we pick, so I count myself fortunate to have a large group of them.

Finally, Thank you Mom, Dad, Uncle (who was an influence in choosing my studies and, helped me throughout my PhD journey), Grandmother and my little sisters for your love, inspiration, sacrifices, efforts and always being there when I needed you.

I want to thank the College of Science and Engineering, NUI Galway, for kindly sponsoring my PhD and housing me while I was pursuing it. Additionally, I would want to express my gratitude to Xperi for providing me with significant funding, supporting me throughout my PhD, and the chance to work on real consumer electronics challenges.

Abstract

The depth estimation problem has made significant progress due to recent improvements in Convolutional Neural Networks (CNN) and the incorporation of traditional methodologies in these deep learning systems. Depth estimation is one of the fundamental computer vision tasks, as it involves the inverse problem of reconstructing the three-dimensional scene structure from two-dimensional projections. Due to the compactness and low cost of monocular cameras, there has been a significant and increasing interest in depth estimation from a single RGB image. Current single-view depth estimation techniques, however, are extremely slow for real-time inference on an embedded platform and are based on fairly large deep neural networks that require a large range of training sets. Due to the difficulties in obtaining dense ground-truth depth at scale across various environments, a range of datasets with distinctive features and biases have developed. This thesis firstly provides a summary of the depth estimation datasets, depth estimation techniques, studies, patterns, difficulties, loss function and opportunities that are present for open research. For effective depth estimation from a single image frame, a method is proposed to generate synthetic high accuracy human facial depth from synthetic 3D face models that enables us to train the CNN models to resolve facial depth estimation challenges. To validate the synthetic facial depth data, a brief comparison analysis of cutting-edge depth estimation algorithms on individual image frames from the generated synthetic dataset is proposed. Following that, two different lightweight encoder-decoder-based neural networks for training on the generated dataset are proposed, and when tested and evaluated across four public datasets, the proposed networks are shown to be computationally efficient and outperform the current state-of-the-art. The proposed lightweight models will allow us to use the low-complexity models, making them suitable for implementation on edge devices. Synthetic human facial depth data can help overcome the lack of real data and can increase the performance of the deep learning methods for depth maps.

List of figures

1.1	Evolution of research in depth estimation. This work classifies the improvements of depth estimation into 3 phases: the early period, the machine learning period, and the deep learning period, in which the depth evaluation methods of monocular images based on deep learning is mainly studied and presented.	2
1.2	The overall diagram of single image depth estimation techniques using DL. These DL techniques are categorized into different models depending on whether the network using GT; single-task and multi-task learning techniques depending on the type of network prediction task.	3
3.1	The general framework and a schematic representation of generating the synthetic human facial dataset	27
3.2	Creating human models in iClone and imported model in Blender - render ground truth process	28
3.3	Sample synthetic RGB images and GT depth images with various variations (head postures, expressions, light variations, camera angles, clothing, views, and backgrounds: plain; textured; real) were generated from the synthetic dataset.	29
4.1	A workflow of the technique for rendering 2D images in Blender	33

List of tables

2.1	Author's Contributions to [1]	19
2.2	Author's Contributions to [2]	20
2.3	Author's Contributions to [3]	22
4.1	Author's Contributions to [4]	33
4.2	Comparison of various depth estimation models with the proposed method	
	FaceDepth	35
4.3	A detailed comparison analysis and properties of the studied methods	35
4.4	Author's Contributions to [5]	36
4.5	Comparison of various depth maps methods with the proposed method	
	LEDDEPTH	38

Nomenclature

Acronyms / Abbreviations

3D	Three Dimensional
AI	Artificial Intelligence
ANN	Artificial Neural Network
AR	Augmented Reality
CE	Consumer Electronics
CE	Consumer Technology
CG	3D Computer Graphics
CNN	Convolutional Neural Networks
CPU	Central Processing Unit
DESI	Depth Estimation Single Image
DL	Deep Learning
DNN	Deep Neural Networks
GANs	Generative Adversarial Networks
GDPR	General Data Protection Regulations
GPU	Graphics Processing Unit
GT	Ground Truth
ML	Machine Learning

SoA State-of-the-Art

VAEs Variational Auto-Encoders

VR Virtual Reality

Chapter 1

Introduction to the Single-Image/Monocular Depth Challenge

1.1 Introduction

Image depth estimation plays a key role in computer vision, which facilitates the understanding and awareness of real 3D situations leading to a variety of applications such as robotic navigation, self-driving, and augmented worlds [6–8]. Active depth approaches typically utilize laser beams, structured illumination, and other reflective surfaces on the object surface to acquire depth point cloud data, explicit surface model construction, and approximate scene depth information [9, 10]. However, producing dense and precise depth maps typically involves relatively significant costs, time of manpower to manually annotate and processing resources [11, 12]. Image-based depth estimation methods mainly contain binocular image depth estimation and single image depth estimation methods. The results of multi view image depth estimation from scenes containing less texture details are significantly uncertain, and require great accuracy in the calibration and triangulation of the acquired images. Depth information retrieved from a monocular image is inconsistent, as different depth information can be projected to a given scene [13, 14].

The progress of image-based depth computation is presented in Fig. 1.1. In the early period, researchers calculated depth maps based on depth cues, including edges [15], focus and defocus [16], and shadow [17]. However, the majority of these algorithms were applied in constrained scenarios [15–17]. With the development of the computer vision, numerous hand-made features and probability-based graph models have been suggested. Such methods

includes scale-invariant feature transform (SIFT) [18], speeded up robust features (SURF) [19], pyramid histogram of oriented gradient (PHOG) [20], Conditional Random Field (CRF) [21], and Markov Random Field (MRF) [22]. These methods were adopted to anticipate multi view and monocular depth maps with parameter and non-parameter learning in the machine learning process [21, 22]. The development of deep learning technology has given tremendous advances to image analysis [23–30], particularly depth estimation.

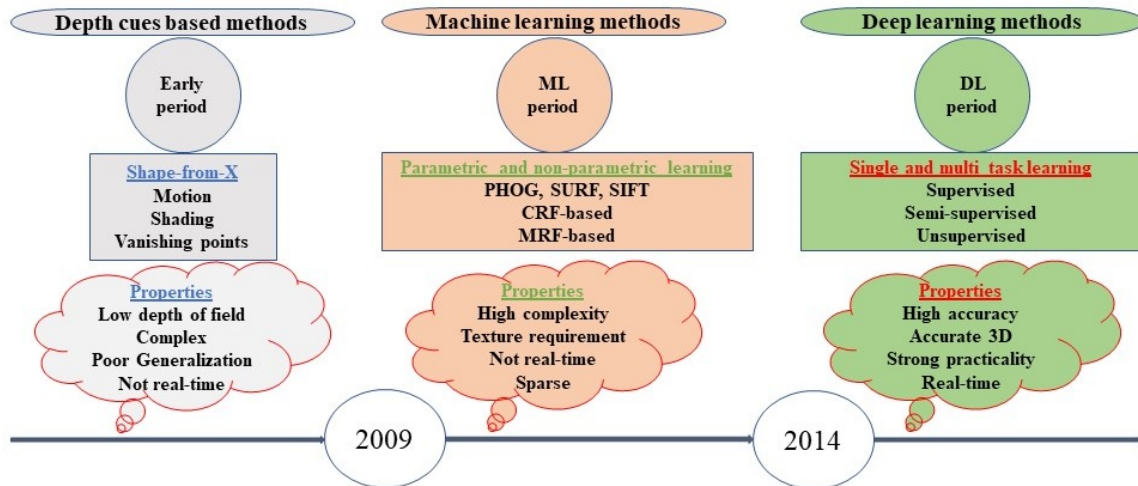


Fig. 1.1 The advancement of depth estimation. This work classifies the evolution of depth estimation into 3 phases: the early period, the machine learning period, and the deep learning period, in which the depth evaluation methods of monocular images based on deep learning is mainly studied and presented [31]

In multi view depth estimation, the disparity of two 2D images (acquired by a binocular camera) is calculated by stereo matching and triangulation to create a depth map [32, 33]. It is difficult to capture enough elements in the image to match when the scene has less or no texture information [34]. As a result, researchers focus has shifted to monocular depth estimation. Monocular depth estimation employs a single camera to acquire an image or video sequence, requiring no additional complex equipment or professional techniques. It has a broad range of application requirements due to the wide availability of only one camera in the majority of real-world applications. As a result, demand for monocular depth estimation has increased in recent years. These approaches perform with a smaller number of operations and have less computational complexity[35]. Researchers have proposed a variety of approaches for monocular depth estimation that can be used in a wide range of applications such as autonomous driving, robotic navigation, and virtual reality [3, 31].

Deep learning techniques for monocular depth estimation can be categorized into three categories: supervised, unsupervised, and semi-supervised training, and single and multi-task

training of depth maps networks. By inferring the scene structures from the GT depth images, the supervised monocular depth maps model estimates the depth maps [1]. Fig. 1.2 shows the overall diagram of deep learning-based monocular depth estimation.

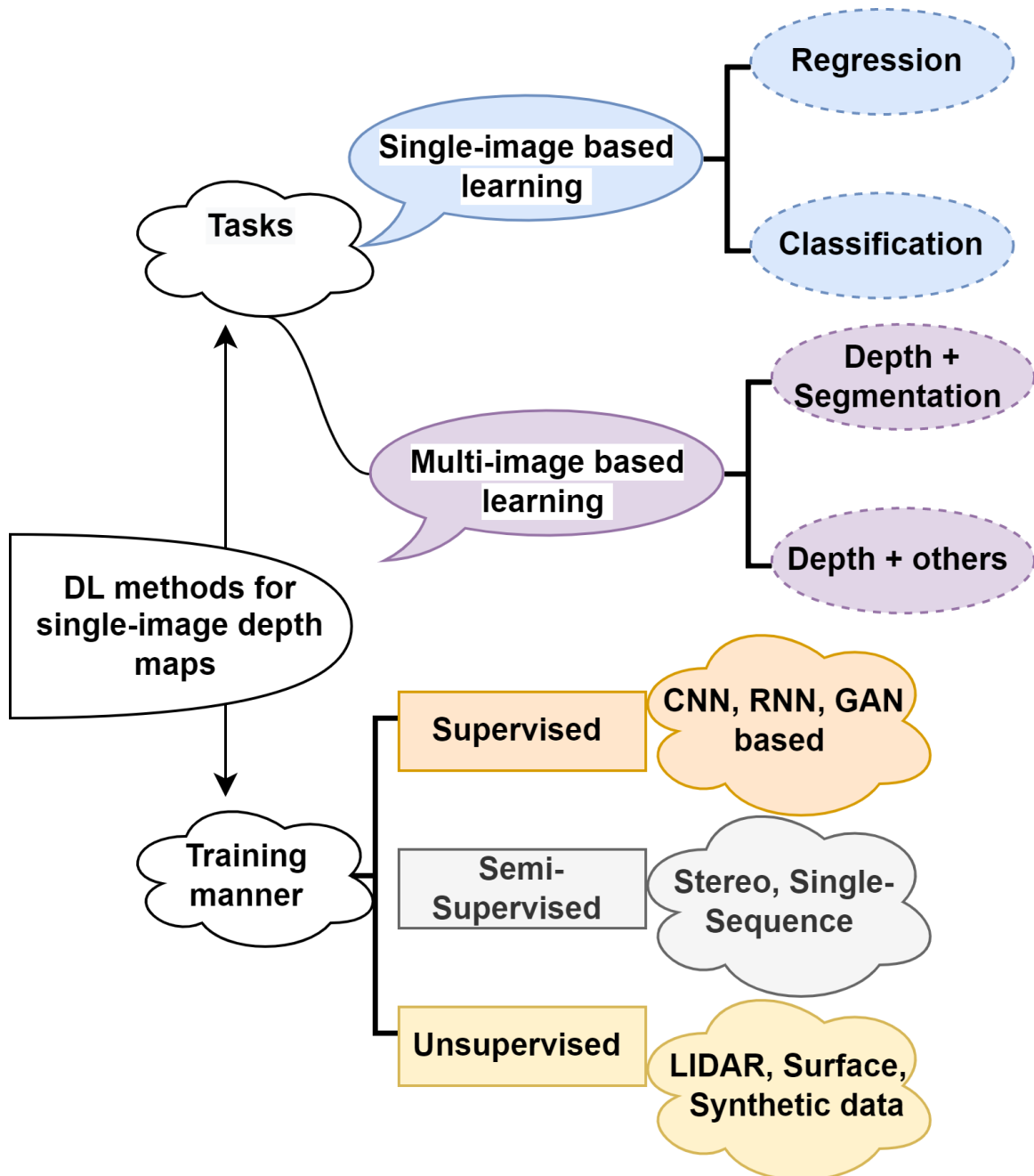


Fig. 1.2 Advances in depth estimation techniques. These DL techniques are categorized into different models depending on whether the network using GT; single-task and multi-task learning techniques depending on the type of network prediction task.

CNN advancements and publically available datasets have recently greatly improved the performance of monocular depth estimation methods [3, 32, 12, 36, 28]. In a scene, CNN can automatically extract spatial information representing depth. It is a kind of feed-forward neural network that simultaneously extract depth information and generate depth maps compared to traditional methods [31]. Convolution operation, pooling layer, fully-connected layers, and activation function are the four key layers that make a CNN. These layers allow CNN to acquire the 2D spatial properties of the input image. The input is transformed into depth features by the CNN layer, the input feature map size is decreased by the pooling layer in the fully connected layer; maximum or average pooling is typically found to output the information at the end of the CNN; and typically, activation function is a continuously differentiable nonlinear function to avoid pure linear combinations. Some typical CNNs are AlexNet [63], VGG [37], GoogLeNet [38], and some lightweight networks, such ResNet [39], DenseNet [40], and such as GhostNet [41], MobileNet [42], ShuffleNet [43]. It serves as the foundation for the current CNN-based systems and it can be used effectively in depth estimation tasks [31].

RNN, a sequence-to-sequence model with memory capabilities, learns temporal information from video sequences. RNN is made up of three units, i.e., an input unit, a hidden unit, and an output unit. The outputs of the previous hidden unit and the current input unit are used as the hidden units input. The spatial features from single image frames are captured by RNN-based supervised learning networks for depth maps [44, 45]. In comparison to CNN-based models, the RNN-based networks encoder is composed with all LSTM (or ConvLSTM) layers, or it combines LSTM and convolutions (ConvLSTM), in order to extract and preserve spatial information for single depth maps.

From the GT depth maps, the supervised depth estimation model can learn the 3D mapping and scale information. GAN [46, 47] was developed by researchers to generate better and more accurate depth maps in comparison to other models, as it is challenging to acquire GT depth maps in real scenes. The depth map is predicted by the generator as a depth estimation network, and the discriminator evaluates whether or not the input depth map is true.

Due to the high expense of collecting GT depth images, some monocular depth estimation networks must be trained with less or no GT in order to reconstruct depth images. These techniques are referred to as semi-supervised or unsupervised learning. Supervised learning methods for monocular depth estimation have the highest accuracy but are highly dependent on GT depth maps. Unsupervised learning methods use geometric limitations of the input images to predict depth maps without supervision, but their accuracy is significantly inferior to supervised and semi-supervised learning techniques, which must overcome scale uncertainty,

obstruction, and other issues. Semi-supervised learning methods that contain information, such as synthetic data, surface normals, and LIDAR, as the semi-supervised learning manners to reduce the network dependence on GT depth maps, which improve the scale consistency and enhance estimated accuracy of depth maps. This is performed in order to efficiently utilize a large amount of relatively inexpensive unlabeled data to improve learning performance.

Deep learning techniques for monocular depth estimation can be divided into two categories based on the different task types. On the one hand, users can train a network exclusively for depth estimation, which is referred to as single-task learning; but on the other hand, researchers can integrate depth estimation with several other related tasks in order to jointly learn for feature representation and enhance depth forecasting accuracy, which is referred to as multi-task learning [31].

Deep learning algorithms have significantly improved 2D face identification, making 3D face recognition more promising. Due to the fact that 3D faces are widely considered to be more discriminative than 2D faces. For 3D face recognition, depth information is the foundation of the majority of deep learning algorithms [48].

1.2 Facial Depth Estimation Challenge

Estimating depth maps from images is a fundamental and crucial problem in CV that could be used in a wide range of tasks such as semantic segmentation, navigation, localization, object detection, mapping, and 3D reconstruction [49–51]. Different techniques can be used to estimate depth, such as stereo vision matching or multi-view, which can estimate the 3D structures of a scene by employing two or more different points of view [52]. It uses two or more cameras to process the scene to determine the disparity maps of the images. Since the cameras in multi-view are calibrated in advance and all the data is contained in the depth maps [53] using geometric constraints methods. A technique commonly used in 3D reconstruction and SLAM, is another way to recover depth maps and the corresponding 3D structures from two or more images [54, 35]. Although, geometry-based algorithms can compute the depth values of sparse features well, they typically rely on image pairings or image sequences. Due to the absence of efficient geometric solutions, it is currently very difficult to create a dense depth maps.

Advanced sensing devices called RGB-D cameras acquire RGB images together with information about the depth of each pixel. These cameras have issues with limited accuracy in range measurement and outdoors sun-lighting sensitivity. Depth sensors, LIDAR, and other sensor-based techniques can also obtain the depth maps information of the images directly [55]. LIDAR is commonly utilized in the automated vehicle sector for depth sensing,

however it can only produce a sparse 3D map. Additionally, the applications of these depth sensors (RGB-D cameras and LIDAR) to robotics applications, such as drones, are limited by their large size and high power requirements [56].

Images of human faces are one of the most common today, and they play a pivotal role in several visual interpretations. Since the facial components and representation in a face image is well-known in human anthropometrics, it is helpful to quantify the distance of a human focus from a single image frame if the camera's field of view is known. When one can directly reconstruct a 3D face model from 2D feature points, the resulting face model lacks characteristics since we are unable to determine the depth of the features from the 2D facial image. In contrast to 2D data, shadow and aesthetic effects are negative for 2D data, but they are advantageous for 3D data since they have rotation invariance and illumination invariance. This indicates that 3D data is reliable and accurate. 3D data have more information than 2D data since they can show richer facial shape features.

The estimation of facial depth and head pose is crucial for the autonomous monitoring of driver concentration in a demanding environment characterized by significant lighting changes, occluded, and high head postures. Using DL algorithms, image-based facial depth estimation has demonstrated encouraging results. However, the field is still in its infancy, and further advancements are anticipated to discuss the difficulties and problems, such as selection and data for training, inferences to dynamic environments, fine-scale depth prediction, reconstruction versus identification. By processing various objects in the existence of occlusions and congested backgrounds, data lack of balance, and how to choose an acceptable objective function and neural model for facial depth estimation.

For an accurate depth map, the depth camera sensor should be capable of faster human-skeletal tracking in addition to being a low-cost camera sensor that outputs both RGB and depth information. This kind of tracking can provide the precise position of human body joints, making comprehensive human behavior investigations easier and quicker. As a consequence, there has been a lot of interest in inferring human faces from depth images and synthesizing depth and RGB images in recent years. Several new facial depths map datasets have been generated in recent years to assist in the confirmation of humanoid facemask action analysis methods. However, the GT of these datasets is relatively low, which can negatively impact the performance of CNN and make them unsuitable for training [4].

The demand for correctly labeled data (GT) is a fundamental limitation of such supervised models, despite the success of DL-based approaches. Specifically for facial depth estimation problems, it is difficult to gather precisely annotated face depth data due to feature variables such as race, age, and gender, as well as environmental influences such as noise, light, and obstruction. Despite that, it is challenging to predict how much data will be required to

train an algorithm. Also the dataset must contain sufficient information about all important classes and edge cases that the algorithm should manage. Each significant dataset requires a tremendous amount of labor and substantial investment, as well as extensive logistical planning which is again very challenging. Also it might take months or even years to compile a large-scale ground truth dataset, which delays DL research and implementations. No assurance can be given that the data gathered will be of a high enough caliber to train the algorithms on all required tasks and use cases.

In addition, several data acquisition metrics, including depth sensing and IMU movement, are susceptible to sensor noise. As a result of uncertain 3D models and camera characteristics, manually annotated key point procedures typically produce erroneous results. The facial depth datasets available include Pandora [57], Eurecom Kinect Face [58], and Biwi Kinect Head Pose [59] captured from real subjects only consists of a small number of images and contains a low GT, thus these datasets are poorly suited for training DL-based depth methods.

Furthermore, the collection of new data from human subjects is now governed by several data privacy protection regulations, such as the GDPR, and is subjected to an increasing amount of severe restrictions making it hard to capture new datasets particularly for faces. It is getting harder and harder to collect ground truth data that includes images or information about living humans.

Another challenge is that implementation of autonomous navigation requires precise depth and 3D data in real-time. Vehicles and battery-powered drones are the two categories of autonomous guidance systems. In these applications, the usage of a camera or cameras is constrained due to interference from varying illumination, reflective surfaces and weather. Utilizing laser scanners to create 3D data for automated driving is possible. Nevertheless, the scanners are costly and involve a substantial amount of power to function, limiting their efficiency on battery-powered drones. The greater difficulty is how to increase the density of the sparse data supplied by laser scanning while maintaining the scene structure.

The improved capability of today's technological devices provides optimism that a perception depth-sensing imaging system will be developed within the next decade or two, and it is hoped that some of the contributions of this study may help in the development of this solutions.

In general, we believe that monocular depth estimation will continue to be developed with a focus on enhancing accuracy, simplicity, and real-time performance. The large number of earlier works primarily concentrate on enhancing depth estimation accuracy through the use of new loss functions or network architectures. Generalization describes how a network performs when used with various cameras, scenarios, and datasets used. There is growing interest in depth networks generalization. The most of the algorithms used today

are trained and tested using the same dataset, which yields excellent results. However, there is frequently significant performance loss as a result of using training and testing sets from various domains or cameras. Utilizing domain adaptation technology and including camera characteristics in depth estimation framework training will considerably increase the depth network's generalisation. Larger networks perform exceptionally well, but their applications face a significant problem because estimation tasks take longer to perform when using deeper networks. The real uses of depth estimation networks will be significantly impacted by their ability to function in real-time on embedded devices. Therefore, the development of lightweight networks based on supervised, semi-supervised learning and pixel-accurate GT is an alternative to improve accuracy while ensuring real-time performance.

The method for creating 3D synthetic human facial models is suggested in this thesis in order to render the appropriate 2D RGB and depth maps along with head pose information. The research community can use this dataset to improve facial depth estimation and apply it to use in real life applications.

1.3 Overview of Contributions in this Thesis

In this section, the accomplishments of this dissertation are briefly summarized. In the subsequent chapter of the thesis, every contribution's associated works is discussed in detail. An introduction paragraph gives a background for the study in each chapter. Then, the objectives of the research are presented, supported by the contribution of the presented research. Finally, a summary of the contributions is presented, analyzing their impact on the entire study field. Furthermore, for each work an introduction, and a table detailing the authors efforts in relation to the four primary criteria described in section 1.6 is provided.

The main contributions of this thesis are categorised into three main areas in the context of the main challenges:

1. General 'review' work on the broad field of single-image/monocular depth maps and SoA NN models; this is captured through 3 review papers throughout the term of the thesis.
2. Contribution to building 2D datasets from 3D models with pixel-accurate depth GT; this is captured in two conference papers, but also in the release of both, 3D dataset and 2D synthetic face-depth dataset; these have DOI numbers on IEE dataports and further validated in the main journal papers.

3. Contribution to improved NN models for monocular facial depth maps, together with methodology for evaluation and, as a sub-contribution, a detailed comparison of SoA models. This is captured in two stand-alone papers.

In the next sections we will give more details on each of these contributions.

1.3.1 Contribution to general Review Work on the broad Field of single-Image Depth Maps and SoA NN Models

1. Khan, Faisal, Saqib Salahuddin, and Hossein Javidnia. "Deep learning-based monocular depth estimation methods—a state-of-the-art review." *Sensors* 20, no. 8 (2020): 2272, [1]. Appendix A contains a copy of the paper published. This is the first comprehensive review of the research in this thesis to depth estimation from single-image frames using DL methods. This study's major goal was to identify alternative network architectures with reduced computational complexity that would be easier to implement in consumer devices even while providing comparable performance to larger CNN architectures. To accomplish this goal, a concise explanation of the monocular depth estimation concept, a problem description, traditional depth estimation techniques, and publically available datasets are presented. Following that, DL architectures for monocular depth estimation were categorized as supervised, self-supervised, and semi-supervised. The SoA methodologies are also thoroughly compared, followed by a discussion and potential future study fields.

2. 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, [2]. The paper published can be found in the Appendix B. This work's major objective is to provide a brief analysis of the depth datasets that are currently available and the loss functions that are applied to the problem domain. The key features and properties of each depth dataset and loss function are explained and compared, and the depth datasets and loss functions are classified into different categories based on use cases. Furthermore, a discussion of challenges and future research as well as suggestions for developing robust depth datasets are presented.

3. Khan, Faisal, Muhammad Ali Farooq, Waseem Shariff, Shubhajit Basak, and Peter Corcoran. "Towards Monocular Neural Facial Depth Estimation: Past, Present, and Future." *IEEE Access* (2022), [3]. This published paper can be found in the Appendix C. This study [3] tries to give all the information that would be needed to do a study on the problems with monocular facial depth estimation. After giving a brief summary of the research on facial

depth maps, facial depth estimation applications, challenges, and how it has been used, a detailed study of publicly available facial depth datasets and commonly used loss functions was given. To help you understand the facial depth map problem better, the important features and qualities of the facial depth dataset are defined and analyzed, and then the loss functions used are described. For each dataset, the dataset description, metadata, ground truth, and relevant data are all listed in a clear way. Also, each loss function is given in a way that allows researchers to choose the best loss function for their needs. It shows and talks about the technical details of neural depth networks and the evaluation matrices that are included in them.

In the second half of the study, a thorough comparison evaluation and, where possible, a direct comparison of facial depth estimation methods are done to lay the groundwork for the suggested model. When tested on four different data sets, the model is better than the current best methods. The unique loss function of the proposed method helps the network learn the areas of the face, which leads to an accurate prediction of depth. The network is trained and tested with real and fake facial images from four facial depth datasets, as well as synthetic images of human faces. A 3D point cloud is reconstructed from the predicted depth maps and compared with SoA methods for 3D reconstructions.

1.3.2 Contribution to Building 2D Datasets from 3D Models with Pixel-Accurate Depth GT

1. The first conference paper: *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, [60]*, a copy of the work published can be accessed in Appendix D. The primary objective of [60] is to generate 3D virtual human models with RGB and depth images. The purpose is to evaluate the validity of synthetic datasets used to solve real-world problems in various application fields. This should enable further methodological enhancements to be made to the generated datasets in order to suit specific use case applications.

2. The second 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, [61]*, Appendix E contains a copy of the work published. The main goal of this research [61] is to create synthetic images of human faces that can be used in a range of environments, such as 3D computer graphics, to simulate real-world problems. This research [61] is an extension

of the previous work to create more variations [60] to make 3D models of faces that look like real ones more variable and robust, along with both 2D RGB and depth images. A full workflow for making a set of synthetic human faces with depth renderings and textured and complex backgrounds. A shallow CNNs-based UNet framework is shown to test the quality of the data in the created datasets. The SSIM loss, gradient loss, and surface normal loss are used to help the network learn the correct depth of the scene and the 3D structure of the face. This is followed by a discussion of the results using a widely accepted evaluation method with five evaluation metrics and showing the experimental results, and implementation details of the trained models on the generated datasets.

These 3D models and 2D datasets with pixel-accurate depth GT and 2D synthetic face-depth images are publicly available and we believe it can significantly help the face depth maps problems. The DOI numbers on IEEE Dataport are as follows:

1. 10.21227/ath9-br59
2. 10.21227/f6zx-bf29

1.3.3 Contribution to improved NN Models for monocular Facial Depth Maps: Training new CNN Models and their Evaluation and, as a Sub-Contribution, a detailed Comparison of SoA Models

1. The first journal paper of this thesis which is main contribution: *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, [4]. Appendix F includes a copy of the work that is published.*

The first part of this paper develops the work of the previous two conference papers [60] and [61] into a comprehensive research study [4] by providing all the details used to generate the datasets and how it is organized.

1. This study presents a system for producing complex synthetic human face datasets by combining multiple variation in synthetic facial data, a synthetic human model with a 3D model design, the iClone Character Creation workflow, and the addition of variations to iClone models.
2. This is followed by the model transfer from iClone to Blender, model manipulation in Blender, construction of 3D scenes in Blender, The Blender camera model, selection of 3D background environments in Blender, ground truth rendering in Blender, and the structuring of the dataset.

3. Using a collection of SoA DESI neural networks, the synthetic human facial depth dataset is first trained and then evaluated.
4. In addition, a new CNN model with a hybrid loss function is constructed, and its performance is compared to that of SoA networks. Initially, SoA DESI algorithms are trained on a synthetic dataset of human facial images, and their performance is compared to that of the proposed network.

2. The second journal paper of this thesis that covers the second main contribution: *Khan, Faisal, Waseem Shariff, Muhammad Ali Farooq, Shubhajit Basak, and Peter Corcoran. "A Robust Light-Weight Fused-Feature Encoder-Decoder Model for Monocular Facial Depth Estimation from Single Images Trained on Synthetic Data." IEEE Access (2023)*, under review [5], a copy of the paper published can be found in the Appendix G. The primary contribution of this study is a new neural facial depth estimation network that predicts accurate facial depth maps from single image frames. This network is substantially smaller and more cost-effective than available SoA facial depth estimation techniques, making it suitable for embedded devices and edge-AI applications. On the basis of an evaluation of four publicly available facial depth datasets, this lightweight network outperforms SoA across multiple major measures. In addition, extensive experiments demonstrate the network's value and generalizability.

1.4 Other Contributions

I have collaborated with my PhD colleagues to incorporate a detailed study about the effective use of head-pose imaging for human head pose estimation. The generation of pixel-perfect synthetic 2D headshot images from high-quality 3D synthetic facial models annotated with precise head poses is suggested. There is also a wide spectrum of age, racial, and gender diversity. A SoA head pose estimation model that has been trained and tested against the widely used evaluation datasets is used to evaluate the dataset. Additionally, a semi-supervised strategy for adapting the visual domain is presented, which trains using both labeled synthetic data and unlabeled real data. Model performance is significantly increased when domain adaptation is used. Additionally, better results than previously published work on this topic are obtained by using a data fusion based transfer learning approach. Chapter 5 provides a summary of these extra contributions in more details.

1.5 List of Publications & Datasets

First Author

Journal Publications

1. **Faisal Khan**; Salahuddin, Saqib; Javidnia, Hossein. 2020. "Deep Learning-Based Monocular Depth Estimation Methods—A State-of-the-Art Review" *Sensors* 20, no. 8: 2272.
2. **Faisal Khan**, Shahid Hussain, Shubhajit Basak, Mohamed Moustafa, and Peter Corcoran (2021). "A Review of Benchmark Datasets and Training Loss Functions in Neural Depth Estimation," in *IEEE Access*, doi: 10.1109/ACCESS.2021.3124978.
3. **Faisal Khan**, Shahid Hussain, Shubhajit Basak, Joseph Lemley, Peter Corcoran. An efficient encoder-decoder model for portrait depth estimation from single images trained on pixel-accurate synthetic data, *Neural Networks*, Volume 142, 2021, Pages 479-491, ISSN 0893-6080, <https://doi.org/10.1016/j.neunet.2021.07.007>.
4. **Khan, Faisal**, Muhammad Ali Farooq, Waseem Shariff, Shubhajit Basak, and Peter Corcoran. "Towards Monocular Neural Facial Depth Estimation: Past, Present, and Future." *IEEE Access* (2022).
5. **Khan, Faisal**, Waseem Shariff, Muhammad Ali Farooq, Shubhajit Basak, and Peter Corcoran. "A Robust Light-Weight Fused-Feature Encoder-Decoder Model for Monocular Facial Depth Estimation from Single Images Trained on Synthetic Data." *IEEE Access*, 2023, under review.

Conference Publications

6. **Faisal Khan**, S. Basak, H. Javidnia, M. Schukat and P. Corcoran, "High-Accuracy Facial Depth Models derived from 3D Synthetic Data," 2020 31st Irish Signals and Systems Conference (ISSC), 2020, pp. 1-5.
7. **Faisal Khan**, Shubhajit Basak and Peter Corcoran, "Accurate 2D Facial Depth Models Derived from a 3D Synthetic Dataset," 2021 IEEE International Conference on Consumer Electronics (ICCE), 2021, pp. 1-6.

Co-Authored

Journal Publications

8. S. Basak, P. Corcoran, **Faisal K**, 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.

Conference Publications

1. Basak S, Javidnia H, **Khan F**, McDonnell R, Schukat M. Methodology for building synthetic datasets with virtual humans. In 2020 31st Irish Signals and Systems Conference (ISSC) 2020 Jun 11 (pp. 1-6). IEEE.
2. Basak S, **Khan F**, McDonnell R, Schukat M. Learning accurate head pose for consumer technology from 3D synthetic data. In 2021 IEEE International Conference on Consumer Electronics (ICCE) 2021 Jan 10 (pp. 1-6). IEEE.

Datasets

1. 3D-Dataset ([62, 63])
2. 2D-Pose Dataset ([62])
3. 2D-Face Depth Dataset ([60, 61, 30])

1.6 Contribution Taxonomy

Due to the fact that this publication-based thesis contains collaborative effort, this section gives an outline of the primary factors that identify primary authorship. The CRediT approach has been adopted by journals in several fields to specify the contributions of individual authors. In the CRediT Taxonomy, all authors' contributions are measured as a percentage point on 14 roles. These are: Conceptualization, Data curation, Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Despite collaborations, most of the work in this thesis is my own; hence, a more compact generalization of this taxonomy that contains the primary criteria has been selected. To be more specific:

1. Research Hypothesis/ Idea.

2. Methodology comprising validation, data creation, formal analysis, instrument selection, software development, implementation, and experiments.
3. Background which includes investigation, formalization, and work done to place the research efforts in a broader context of literature in a given field; this may include some aspects of writing (literature reviews) and informs aspects of project administration and supervision, as well as ensuring that the methodology employed is typical of that used in the area of publication.
4. Manuscript preparation which includes all aspects of writing manuscript preparation including Writing – original draft, Writing – review & editing, and Visualization except those specified in the next criteria.

Chapter 2

Contribution to General 'Review' Work on the Broad Field of Single-Image Depth Maps and Neural Models

Introduction

The major topics of this chapter include the broad field of single-image depth map research, applications of depth maps, the SoA DL techniques employed and their performance, evaluation matrices, depth datasets, and loss functions. It provides a thorough and organized survey of SoA deep learning-based monocular depth estimation algorithms. The review purpose is to help the reader understand this developing area, which has attracted the interest of the computer vision community in recent years. Furthermore, it makes recommendations on how to generate new depth datasets and what variables can make the datasets significant for DL methods. A brief discussion of which method, dataset, and loss function should be chosen for a particular depth mapping problem, in particular facial depth estimation, follows. A detailed analysis of SoA methods, publicly accessible depth datasets, and commonly used loss functions is given for consideration after providing a brief overview of the research on depth maps and how it has been used. This is done to help the research community better understand the depth map problem and to benefit from speeding up their research and use of it for practical use case applications.

Three review papers written during the course of the thesis and the main contribution of each paper is listed in the following upcoming section.

2.1 Research Objectives

Monocular depth estimation is a key problem in computer vision, with applications in robotics, scene perception, 3D reconstruction, and medical imaging [64–67]. Since there are no good measures for recognizing depth from a single image, this problem remains challenging. Such images, for example, missing temporal information and stereo correspondences. Traditional depth estimation methods [68, 69] rely primarily on multi-view geometry, such as stereo images. The majority of binocular or multi-view approaches can estimate depth details with reasonable accuracy. However, for many applications, their processing time and memory needs are serious barriers [36].

The idea of using a monocular image to gather depth details has potential to overcome the memory issue, however capturing global features of a scene like texture variation or defocus information is computationally challenging. Convolutional Neural Networks (CNN) and publicly available datasets have recently improved the performance of monocular depth estimation algorithms dramatically [70–74].

It is vital that researchers in this field are made aware of the large range of publicly available depth datasets as well as the attributes of various depth estimation loss functions. The right training data, along with the suitable loss functions, will speed up new research and allow for more accurate comparisons with the SoA methods. A loss function is a metric that measures how well a prediction model predicts the expected result. The learning problem is transformed into an optimization problem, a loss function is defined, and the method is optimized to minimize the loss function.

Datasets are the building blocks for analyzing the performance and validating the results of artificial intelligence models, and they play a critical role in scientific research. Data captured in various environments (e.g., indoor vs. outdoor scenes), of various 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 are all included in different datasets. Every dataset has its unique set of characteristics, as well as difficulties and biases [75]. Large dataset collections gathered via the internet have a number of difficulties, including image quality, accuracy, and unknown camera characteristics [76, 77].

High-quality datasets can aid researchers in the development of depth solutions for specific computer vision depth challenges [78, 79]. Indoor/outdoor, portrait/driver, half/full body scene, indoor small room, large street scene, large indoor scene, landscape/cityscape, and medical are some of the different types of depth datasets. Depth data is a map of per-pixel data that contains depth-related information. To aid rendering and computer vision applications, a depth data object contains a disparity or depth map as well as conversion

algorithms, focus information, and camera calibration data. To improve generalization, researchers should mix several datasets during training, validation, and testing, however caution is required when merging datasets with different features [2].

Loss functions are how the algorithm fits data in the first place, thus they provide more than simply a static description of how your model is behaving. In the process of optimizing or identifying the best weights for given data, most ML algorithms utilize some type of loss function. Importantly, the loss function that chooses is directly related to the activation function that has to choose the neural networks output layer. These two design concepts are connected together. Considering the output layer configuration to be a choice regarding the framework of the prediction problem, and the loss function selection to be the method for calculating the error for a given model of the problem. The loss function calculates the difference between the network GT and the estimated output, which is used to adjust the deep network parameters. This is accomplished by backpropagating the loss function error to the first layer of the training process, modifying the network weights at each iteration.

2.2 Summary and Discussions of Contributions

In this chapter, three journal review papers are proposed to cover the main field of single-image depth maps, datasets for training and validation purposes of DL architectures and loss function used for depth particularly for facial depth maps. The following contributions of the proposed works are presented.

2.2.1 Deep Learning-based Monocular Depth Estimation Methods—a State-of-the-art Review

The first review paper: *Khan, Faisal, Saqib Salahuddin, and Hossein Javidnia. "Deep learning-based monocular depth estimation methods—a state-of-the-art review." Sensors 20, no. 8 (2020): 2272, [1], see Appendix A.*

Peter Corcoran is not listed as a co-author of this work since he organized a special issue and requested the corresponding author Hossein Javidnia to submit a manuscript for it in order to ensure an unbiased review process. The author's contributions to the four major criteria, as explained in section 1.6, for the research works [1], are presented in Table 2.1.

This is the first brief literature review of the research work to understand the estimation of single-image depth maps using DL methods. For use case applications, monocular depth estimation algorithms must be high-performing and robust. Furthermore, existing depth estimation solutions are properly evaluated in terms of the degree of supervision, accuracy,

Table 2.1 Author's
Contributions to [1]

Contribution Criteria	Contribution Percent
Hypothesis/Idea for Research	FK 70%, HJ 30%
Experiments and Evaluations	FK 90%, HJ 10%
Background	FK 90%, HJ 10%
Preparation of the Manuscript	FK 70%, HJ 20%, SS 10%

depth range, computation time, and memory requirements for deployment in consumer devices such as robotics, AR/VR headsets, and autonomous vehicles. This study looks to find a potential network architectures with less complexity to make it easier to deploy such solutions in consumer devices while maintaining performance and remaining competitive when compared to larger CNN architectures.

A brief explanation and basic concept of monocular depth estimation, problem description, classic depth estimation methods, and publicly available datasets are presented to achieve the study objectives. Following that, supervised, self-supervised, and semi-supervised deep learning architectures for monocular depth estimation are briefly discussed. A comprehensive comparison of the SoA techniques is also presented, followed by a discussion and possible future research areas. The main contribution can be further explained in detail as follows:

1. A number of important datasets that are particularly well-suited to the depth estimation problem are presented. They include images and corresponding depth maps from various perspectives, highlighting the most commonly used datasets for scene analysis. GT depth images for datasets are frequently captured with consumer-level sensors like the kinect and velodyne laser scanner. A summary is given (Appendix A, section 2.3). In addition, the datasets main properties such as labelled images information, annotation, size and captured scenes details are studied.
2. A CNN trained on RGB-images and the corresponding depth maps is used in the majority of deep learning-based approaches. These techniques are divided into three categories: supervised, semi-supervised, and self-supervised. All of the categorization methods, network structures, as well as the training procedure and their primary loss functions, are briefly presented (Appendix A, section 3).
3. The most commonly used quantitative metrics for evaluating the performance of monocular depth estimation methods are presented (Appendix A, section 4).
4. The methods are briefly evaluated and the results are compared in terms of the performances matrices. These models inference time, parameter count, depth accuracy,

memory usage, and training environment are all evaluated. A visual comparison of the five SoA depth map approaches is provided, with sharper boundaries and a lower relative scale in (Appendix A, section 4, Table 4-6).

5. Color maps appear in a variety of computer vision and ML applications, ranging from showing depth images to more conceptual uses such as image differencing. Colorizing images makes it easier for the human visual system to pick out information, evaluate numeric measures, and spot patterns in data. Jet is a high contrast color mapping method that is widely used in computer vision applications. It is useful for highlighting even weakly distinguishable image features that is used. There is some color deviation in the images, which is due to the low GT depth images and missing depth pixel values, causing it to appear in the wrong direction (Appendix A, section 4, Fig. 1).

By showing the importance of DL-based monocular depth estimation methods, cameras may be able to compete as a reliable source of 3D data and have the potential to be optimized for deployment on smart and consumer platforms.

2.2.2 A Review of Benchmark Datasets and Training Loss Functions in Neural Depth Estimation

Based on the results of this study, we expanded the scope of our research to examine in-depth information about depth datasets and widely-used loss functions for depth estimation which is the second review paper of this chapter.

The second review paper: *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, [2]*, a copy of the paper can be found in the Appendix B.

Table 2.2 shows the author's contributions to the four important criteria, as defined in section 1.6, for the research works [2].

The published datasets show significant variation in terms of size (ranging from 5 to >1,800

Table 2.2 Author's
 Contributions to [2]

Contribution Criteria	Contribution Percent
Hypothesis/Idea for Research	FK 70%, SH 20%, SB 10%
Experiments and Evaluations	FK 70%, SH 10%, SB 20%
Background	FK 60%, SH 10%, SB 10%, MM 10%, PC 10%
Preparation of the Manuscript	FK 60%, SH 10%, SB 10%, MM 10%, PC 10%

classes), sensors used, image quality, and other factors. For this diversity, a number of datasets are available for many researchers, but it is not always easy for researchers to choose the best dataset. This study aids researchers in finding the correct dataset and loss function for depth estimation tasks. Following a brief description (literature, concepts), datasets are assessed in terms of citations, and depth datasets are classified based on their depth applications. Each depth dataset's key features and characteristics are described and analyzed. After that, depth-based loss functions and a depth dataset mixing technique are thoroughly discussed. Finally, reviews of cutting-edge deep learning-based depth estimation algorithms, discussions on problems and future research, and suggestions for developing comprehensive depth datasets are offered. The following are the important aspects of this paper:

1. Based on their use, these depth datasets are divided into five categories: (i) people identification and action recognition, (ii) faces and facial position, (iii) perception-based navigation (i.e., street signs, roads), (iv) object and scene recognition, and (v) medical applications. Each depth dataset's key characteristics and properties are discussed and compared (Appendix B, section 3, 4, and 5).
2. In order to generalize model results across multiple contexts and application situations, a mixing technique for depth datasets is provided (Appendix B, section 4).
3. Three of the most popular datasets are evaluated using state-of-the-art deep learning-based depth estimation algorithms (Appendix B, section 6).
4. Also mentioned are depth estimation loss functions that can benefit in the training of deep learning depth estimation models across a variety of datasets (Appendix B, section 7).
5. Finally, a discussion of problems and future research, as well as recommendations for developing comprehensive depth datasets, is provided to assist researchers in producing diverse and useful depth map datasets (Appendix B, section 8).

The generalization ability and robustness of the DL model is heavily influenced by the quality of the datasets and a suitable loss function. More data of higher quality, more scene categories and a proper loss function are required to improve depth estimation tasks performances.

2.2.3 Towards Monocular Neural Facial Depth Estimation: Past, Present, and Future

We have further extended the research to image-based facial depth estimation using DL algorithms and provide all of the necessary information for carrying out a study on the

problems related to monocular facial depth estimation while keeping in mind the significance of depth datasets and loss functions used for DL methods which our third review paper of this chapter.

This third paper: *Khan, Faisal, Muhammad Ali Farooq, Waseem Shariff, Shubhajit Basak, and Peter Corcoran. "Towards Monocular Neural Facial Depth Estimation: Past, Present, and Future." IEEE Access (2022), [3], is published and a copy of the paper can be found in Appendix C.*

Table 2.3 shows the author’s contributions for the research works [3].

This paper [3] provides a brief overview of the literature and applications of facial depth

Table 2.3 Author’s
 Contributions to [3]

Contribution Criteria	Contribution Percent
Hypothesis/Idea for Research	FK 70%, MAF 20%, WS 10%
Experiments and Evaluations	FK 70%, MAF 10%, WS 10%, SB 20%
Background	FK 60%, MAF 10%, WS 10%, SB 10%, PC 10%
Preparation of the Manuscript	FK 60%, MAF 10%, WS 10%, SB 10%, PC 10%

map research, a detailed study of publicly available facial depth datasets and commonly used loss functions used for DL methods to estimate the face depth estimation. This work differs from the prior work [2], since it is based on facial depth map tasks. The primary takeaways are as follows:

1. To help better grasping the facial depth map problem, a brief literature review and applications of facial depth map method research are offered, and the important characteristics and qualities of the facial depth dataset are defined and analyzed, followed by the loss functions employed. The dataset description, metadata, ground truth, and pertinent data for each dataset are listed systematically (year of publication, ground truth information, image size, kind, objects per image, and multiple images) (Appendix C, Sections 1, 2, and 3).
2. Additionally, each loss function is provided in a way that enables the research community to select the optimal loss function for their task in facial depth estimation (Appendix C, Section 3, Subsection B).
3. The technical details of neural depth networks, as well as the associated evaluation matrices, are presented and discussed (Appendix C, Section 4).
4. The second half of the study includes training method, results discussion, a complete comparison evaluation and, where possible, a direct comparison of the trained facial

depth estimation model that was trained on the synthetic facial depth dataset (Appendix C, Sections 5, 6). When tested across four datasets, the model surpasses current SoA approaches. The proposed method unique loss function aids the network in learning the facial areas, resulting in an accurate depth prediction. The network is trained and tested using synthetic human facial depth datasets and real and synthetic facial images from four facial depth datasets are used for evaluation and testing (Appendix C, Section 7).

Naturally, synthetic facial data will not have the same richness in terms of skin features as real image data. However, due to the small number of images in real-world datasets with low quality GT information, this benefits using synthetic data to train a neural network to achieve equivalent or better accuracy to SoA models trained on real-world data. Based on the synthetic human facial depth dataset, the model was trained and tested on four different datasets. The results show that it is better than MiDaS, DPT, and BTS. Researchers need to know that real facial depth datasets like Pandora, Eurecom Kinect Face, and Biwi Kinect Head Pose do not work well with the generalization performance of the models that were studied. In addition, most depth GT is prone to errors because of practical limitations on how much data can be collected. There are a number of difficulties in the depth of GT data in these datasets, which makes it hard for models to learn about facial depth maps.

It makes sense to think that using a scalable loss function and training method helps you learn more about facial depth and accuracy. The point clouds can be generated from a different angle that provides more details of the 2D scene by using the model input RGB images and the network predicted depth maps. Depth also allows you to use computational photography features like autofocus and portrait mode in high-end phones, which are especially useful at night when depth is difficult to obtain with traditional cameras but easy to obtain with a LiDAR.

The next chapter 3 will further extend the research work to the second contribution of the thesis and will briefly explain how to generate 2D datasets from 3D models with pixel-accurate depth GT.

Chapter 3

Contribution to Building 2D Datasets from 3D Models with Pixel-accurate Depth GT

It is well recognized that data gathering, availability, and preparation are the most significant bottlenecks in ML/DL pipelines, as illustrated by the studies detailed in Chapter 2. Neither of the datasets currently available is sufficiently robust to permit the training of a model that performs well on real images of a variety of circumstances. We currently have a range of datasets that could be complementary to one another but are each biased and lacking in some important information [3]. Synthetic data is less expensive than real-world data and has the potential to provide more accurate GT. This chapter focuses on the synthetic human facial data generated from 3D models using computer graphic open-access software iClone and Blender.

As the head models, camera parameters and positions, scene illuminations, and other constraints can be controlled within the 3D environment, creating synthetic facial images using computer graphics software offers an affordable and sufficient amount of accurately labeled data with comparatively little effort and complexity.

3.1 Synthetic Data and Tools

One of the significant issues in modern AI is a lack of datasets, as available datasets are often too small to train DNN models. When such data is captured without a label, the manual labeling task is time-consuming, costly, and prone to human error. Simple and open source advanced 3D tools, including iClone [80] and Blender [81] can be used to create 3D

models and render 2D RGB and GT images. They have a wide range of poses, hairstyles, expressions, and structures, and their 2D appearance is influenced by external factors such as lighting and camera location. It is possible to generate a number of synthetic data required to train CNN models using these tools. Rendering synthetic human face images would be extremely effective for a range of tasks since, it provides sufficient realism to generate different ground truth. To enrich the datasets, complex backgrounds, depth, movement, body-part edge detection, camera and light orientation can be generated.

3.1.1 The Challenges of ‘Real-World’ Data

Data acquisition, accessibility, and preparations are well-known constraints in ML/DL pipelines [82]. An optimal dataset would include all possible sensing and environmental conditions, but because it is difficult to collect data for all possible cases, real-world datasets are sparser, error prone, and time consuming. The most of currently available datasets have rather low accuracy GT, making them unsuitable for training DL models. A potential method for gathering a significant amount of depth data is the synthetic data produced by the open source 3D graphics engine tools. In order to improve single image depth estimation, researchers have created synthetic datasets with accurate depth GT. During training, it is challenging to bridge the domain gap between synthetic and real datasets [83]. Adversarial learning and style transfer methods including GAN can be used to predict depth maps of real scenes [84] in which the results are depended on trained models with a large number of data with GT depth maps. The trained network can be trained on both real and synthetic data using the domain adaptation approach before being immediately deployed during the test stage to predict depth maps from real RGB images for better generalization. Furthermore, due to practical constraints in data acquisition, most of the depth GT have less variations and number of images. Datasets with various face pose representations are particularly vulnerable to inaccuracies in depth GT data [26].

Additionally, the collection of facial data from persons is now governed by several privacy laws and ethical constraints. The GDPR governs the acquiring and disclosure of personal information data in Europe, posing additional constraints for researchers with live human information [85, 86]. This offers a case for creating low-cost synthetic datasets with minimal complexity and a large amount of labelled data that resemble actual human model elements such as camera settings, positions, lighting locations, scene illumination conditions, and other limitations in a 3D environment. As data complexity increases, numerous data quality challenges eventually arise, which could restrict use case and applications [?]. For instance, it is also difficult to identify biases in data and other underlying quality problems, which

limits the results obtained from such data. High-quality data sets are more important than ever to provide reliable analyses and insights.

3.2 Research Objectives

The fundamental contribution of this chapter is to present a detailed framework for constructing synthetic 3D human face models with the corresponding ground truth depth data. Obviously, synthetic facial data will lack the richness of skin features found in real image data. However, considering the numerous advantages of utilizing synthetic data to train a DL model, a crucial question that we answer is whether we can achieve reasonable performance accuracy of SoA DESI models trained on real-world data.

The general framework of this chapter is illustrated in Fig 3.1, which also includes a step-by-step process description of generating the synthetic human facial dataset. This work consists of two papers and it evolved over an 18-24 month period. In Fig 3.2, which includes a step-by-step explanation of developing the synthetic human facial models, the entire procedure of making 3D models in iClone and creating variations, then importing them into Blender is shown. Shubhajit Basak was the main driver of this project, with some support from me throughout period.

The first research study: *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, [60], published and a copy of the work can be accessed in Appendix D. The "Realistic Human 100" models in the iClone software are used to produce virtual human models based on the subsequent procedures:*

1. The original characters of the virtual human faces are made using the iClone character creator.
2. Character creator virtual human face models are imported into iClone. Furthermore, different expressions such as neutral, angry, happy, sad, and scared are added in iClone to the face models to introduce variations.
3. The created virtual human face models are exported in FBX format from iClone to Blender, which provides appropriate rigging and is used for exchanging the 3D information.

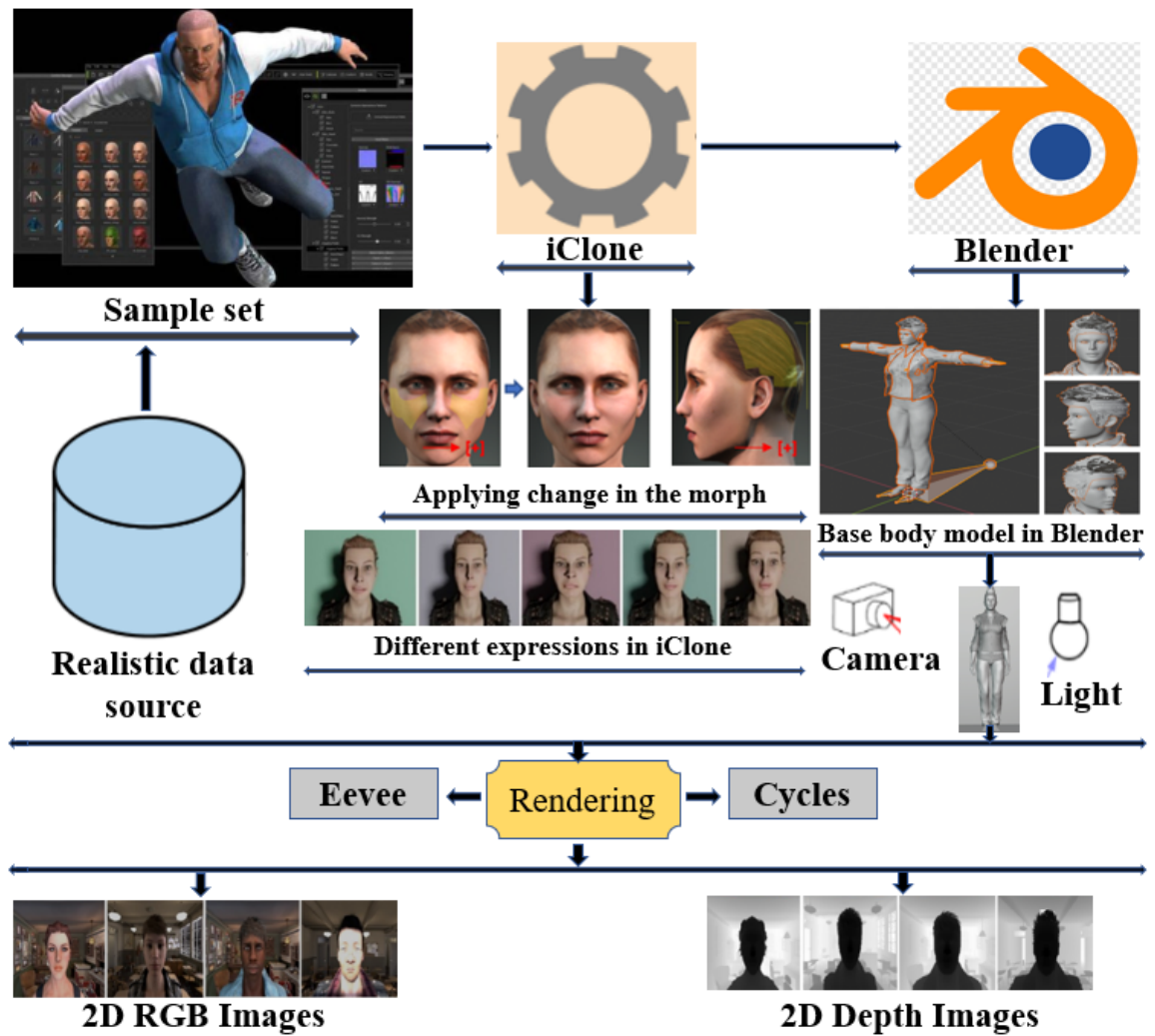


Fig. 3.1 The general framework and a schematic representation of generating the synthetic human facial dataset

4. The cameras and lights are fixed in place, and the models corresponding distances are varied between 700 and 1000 mm. The focal length and sensor size are both set to 60mm and 36mm. In the virtual scenes, the facial models are rotated in Blender.
5. The cameras near and far clip are set to 0.01 and 5 meters, respectively, to generate RGB and depth images of faces in a wide range of positions. The facial models are rendered with a resolution of 480X640 on a static background image.
6. Render passes are configured in Blender to generate the synthetic facial RGB and ground truth depth images. The branched path tracing method is used to reduce noise generated during the rendering process.

- The images are rendered in the perspective view using the Cycles engine and Eevee to obtain RGB images with corresponding facial depth using Python plugin scripts.

Fig. 3.2. Shows creating the models and complete procedure of the datasets generation. Virtual human models are created in iClone software using the Realistic Human 100 models and then imported to Blender to further generate the RGB and depth images (Appendix D, Section III). This is followed by details on the analysis of two SoA CNNs for estimating facial depth, as well as a conclusion and future research (Appendix D, Section IV and V). The

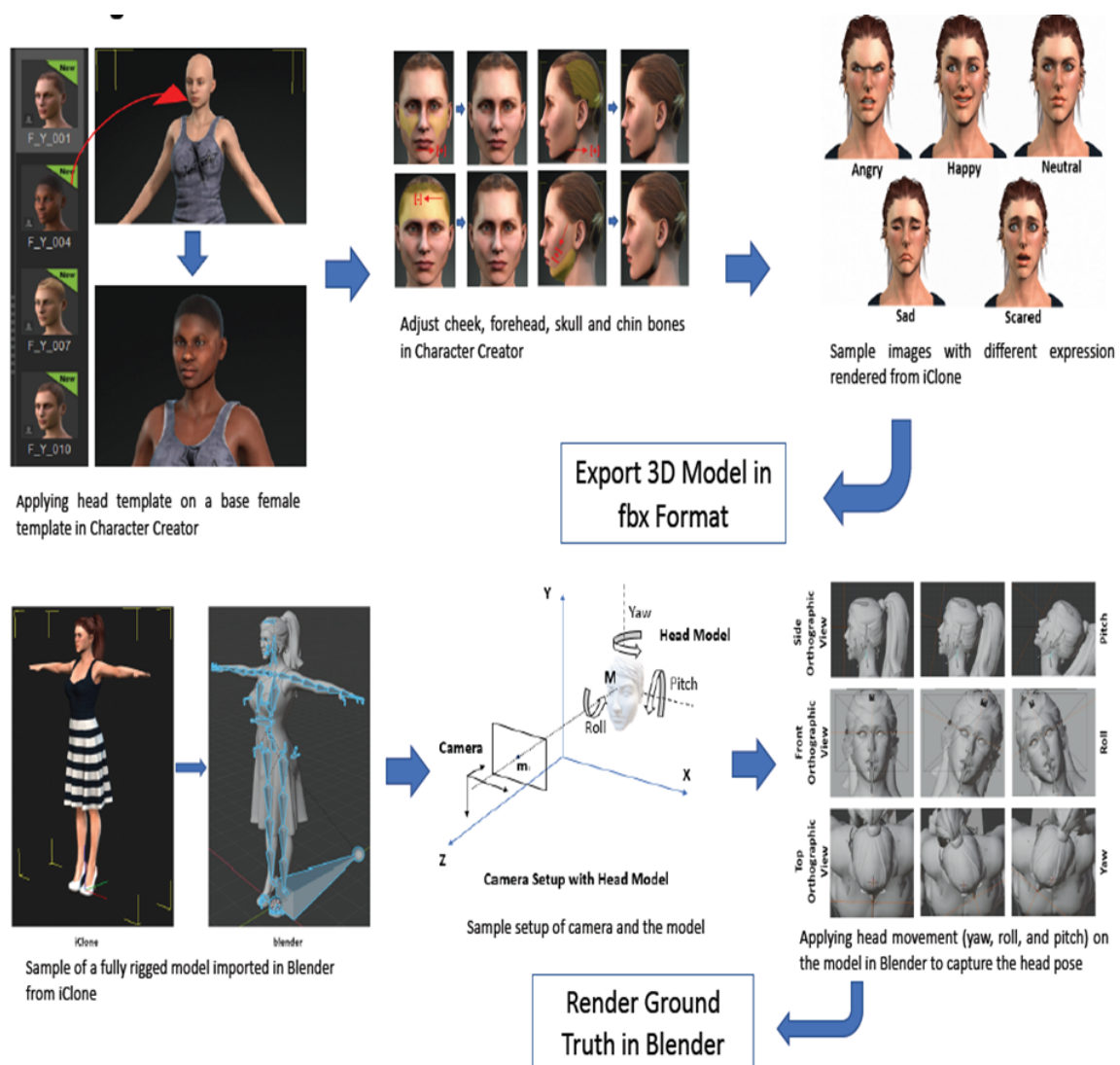


Fig. 3.2 Creating human models in iClone and imported model in Blender - render ground truth process

second 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*, [61], Appendix E contains a copy of the work published. The following steps are used to further generalize the previous work [60]:

1. The position of the camera is changed at various points to the human facial models with the associated ground truth depth.
2. While the camera parameters are set by changing the field of view (FOV), the clip zoom in-out values, the sensors size, the depth of focus, and the f-stop values, the rigs animations are controlled by constraint keyframes and shape keys.
3. The translations and rotations of the neck bones are transferred to the arbitrary object while the boundaries of the original object are maintained.
4. To offer variation to the background, a mix of plain, textured, and real images was used. The scenes background was changed to create more variation in order to increase model generalization. The complex background was created using the Blender eevee classroom and barbershop scenes.
5. To evaluate and compare dataset quality, a shallow autoencoder with skip connection-based UNet architecture is proposed and trained, evaluated, and tested against SoA methods.

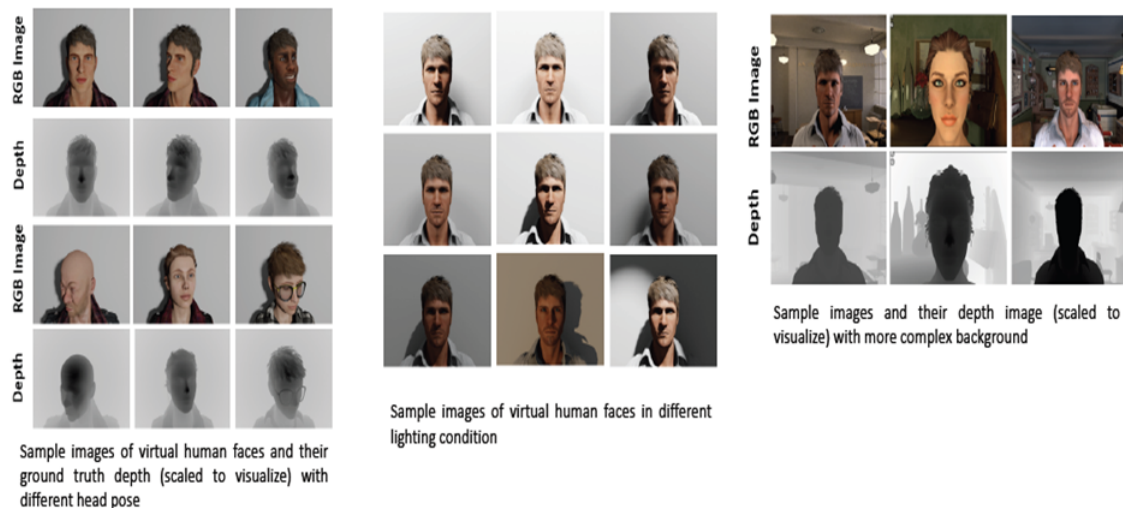


Fig. 3.3 Sample synthetic RGB images and GT depth images with various variations (head postures, expressions, light variations, camera angles, clothing, views, and backgrounds: plain; textured; real) were generated from the synthetic dataset

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. 3.3. (Appendix E, Section III) provide a comprehensive workflow for building a synthetic human facial dataset with ground truth depth rendering, textured and complex backgrounds. To validate the created datasets data quality, a shallow UNet architecture is presented (Appendix E, Section IV-A). The SSIM loss, gradient loss, and surface normal loss are used to assist the network in learning the correct depth of the scene as well as the 3D structure of the face, followed by a discussion of the results, implementation details of the trained models on the generated datasets (Appendix E, Section IV-B- Section IV-E).

3.3 Summary and Discussions of Contributions

One of the primary objectives of [60] is to create 3D virtual human models with corresponding RGB and depth images. This should help researchers to train CNN models to solve real-life challenges in different domains of application.

The generated synthetic dataset, which has a total size of 650GB is divided into two folders. It consists of 2D rendered RGB and GT depth images and 3D virtual human models. The 3D virtual models folder, which is further separated into sub-folders, contains all of the CC and iClone data information (textures,.fbx,.fbm, and. blend) for each subject (male, female). The 2D-generated images folder contains 56 and 44 subjects, respectively, in the male and female sub-folders. The three various sorts of backgrounds—simple, textured, and complex—are kept in three different routes for these subjects. Each of the five primary folders in the sample and texture path—happy, sad, neutral, afraid, and angry—contains the RGB images, depth images, and raw head posture information for every frame. The sample and textured folders' structure of the organization are shared by the classroom and barbershop main folders, which make up the complex directory. Our synthetic dataset is available for no charge download and it can be used in scientific research studies.

3.3.1 The Future of Synthetic Data

The use of synthetic data has increased dramatically during the last decade. While it saves corporations time and expenses. It lacks outliers, which occur spontaneously in real data and are critical for the accuracy of some models. It is also important to note that the performance of the synthetic data is frequently dependent on the data used for production. Biases in the input data can simply spread into the synthetic data, affecting the model training and evaluation results. Using high-quality synthetic data can be more effective than

using available real-world datasets, which give us comparable or better accuracy. Finally, it necessitates further output control, specifically comparing the synthetic data with human-annotated real data to verify that inconsistencies are not created. Despite difficulties, synthetic data remains a promising field. It helps us to create novel AI solutions even when real-world data is unavailable. Most significantly, it enables enterprises to create products which are more inclusive and realistic of their based-on consumers diversity.

The research will be expanded upon in more detail in the following chapter by developing, training, evaluating, and testing SoA DESI approaches. Also, as well as comparing their performance on real datasets, a synthetic human facial depth dataset that has been produced will be utilized.

Chapter 4

Contributions to Improving the Accuracy of Facial Depth Estimation

This chapter will focus on the major contributions of this thesis. The contributions are presented in two journal research papers, and they are further explained in the following subsections.

4.1 Deep Learning Model for Portrait Depth Estimation from Single Images trained on Pixel-Accurate Synthetic Data

This section expands on the work of the previous two conference papers [60, 61], transforming it into a detailed research paper: *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, [4]*, Appendix F includes a copy of the work that is published. Contributions of the author to the four key criteria, as defined in the section 1.6 for the research works [4], are summarized in the table 4.1.

4.1.1 Research Objectives

As a preliminary step towards evaluating the work [60], [61], it is required to construct and comprehend a brief pipeline that explains in full the unique synthetic data samples of human faces and their related depth maps. A brief description of how the 3D models are generated is presented, followed by a demonstration of how the 2D RGB and corresponding GT depth

Table 4.1 Author’s

Contributions to [4]

Contribution Criteria	Contribution Percent
Hypothesis/Idea for Research	FK 70%, SB 20%, PC 10%
Experiments and Evaluations	FK 70%, SH 20, JL 10%, PC 10
Background	FK 70%, SH 10%, SB 10%, PC 10%
Preparation of the Manuscript	FK 70%, SH 10%, JL 10%, PC 10%

are rendered, as well as how the dataset is organized in (Appendix F, Section 3). Fig. 4.1 illustrates the technique for rendering 2D images in Blender. Using current facial depth

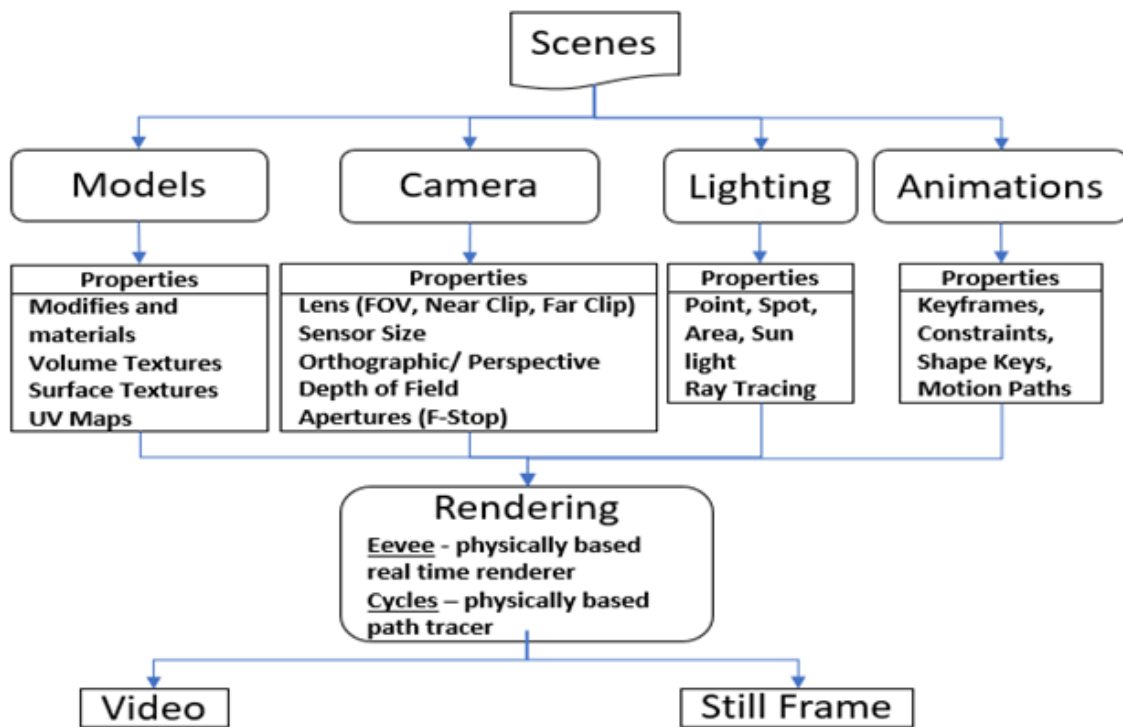


Fig. 4.1 A workflow of the technique for rendering 2D images in Blender

datasets, the majority of them include very poor GT, making them inappropriate for training DL models [87], [88], [89]. Furthermore, due to practical constraints on data collection, most datasets are prone to inaccuracy. Multiple face depth datasets are particularly prone to depth GT data errors and off axis faces will not capture the nuances of facial features. Following that, there was a need for pixel-accurate face depth datasets that could be trained by utilizing DL methods with more samples, identities, and image variations, as well as the correct depth GT, which motivated this sections research effort. As a result, DL approaches can increase

performance and be applied in real-world applications. The main questions that were the foundation for this work motivation:

1. Can we train a more accurate depth model with synthetic data?
2. Does it work correctly on real-data?
3. How accurate models trained on real data are compared to those trained on synthetic datasets?

An improved lightweight encoder-decoder model is proposed and its performance is evaluated and compared against the existing SoA methods for facial depth estimation on four real depth datasets. The lightweight encoder-decoder network consists of input, output images and a two-stage mechanism in the network. The encoder-decoder learns to map datapoints from an input domain to an output domain. The encoder function compresses the input into a latent space representation in the first stage, while the decoder function predicts the result in the second stage. To factorize the CNN layers into depthwise and pointwise layers in the encoder, a MobileNet pre-trained network used which is based on the depthwise decomposition technique. The filtration function, which collects low resolution information from the input image, is used by each of the depthwise layers.

This paper also offers comprehensive comparison of various depth estimation models with the proposed methods FaceDepth, LedDepth, BTS, Densedepth and UNet-simple with various base models (EfficientNet-B0, EfficientNet-B7, ResNet-101, ResNet-50, DenseNet-201, DenseNet-161). The main properties such as learning rate selection, computational complexity, optimizers, number of parameters, input/output size resolution are studied and analyzed.

In terms of accuracy and depth range, based on the evaluations the proposed method achieved the best performance as compared to other SoA methods, BTS [90], Densedepth [91] and UNet-simple [92] with various base models (EfficientNet-B0, EfficientNet-B7, ResNet-101, ResNet-50, DenseNet-201, DenseNet-161) shown in Table 4.2. The evaluation matrices are defined in (Appendix F, Section 6.) Calculating the RMSE, a metric that indicates the average distance between the predicted values from the model and the actual values in the dataset, is one technique to determine how well a model fits a dataset. The better a particular model fits a dataset, the lower the RMSE. RMSElog measures the amount of divergence of predicted probability with the real GT. SqRel divides the total squared error of the predicted output by the total squared error of the Gt to normalize the total squared error of the predicted output. AbsRel is the relative absolute difference, because the mean difference is divided by the arithmetic mean between the predicted and GT values.

On the synthetic human facial dataset, the proposed network achieved 0.0105 RMSE and threshold accuracy of 0.9996 with $\delta < 1.25^3$ as shown in Table. 4.2 (row 16). The

Table 4.2 Comparison of various depth estimation models with the proposed method FaceDepth

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

proposed method is shown to have a significantly reduced memory footprint with improved computational efficiency as compared to other SoA methods, [90], [91], UNet-simple [92] as shown in Table. 4.3 (row 6). At 16.41 G-MACs per frame, this approach detailed analysis can be found in Appendix F. The created synthetic human facial depth dataset is analyzed using the SoA DESI neural networks (Appendix F, Section 4). Initially, SoA DESI techniques are trained on a synthetic human facial dataset, and their performance is compared to that of the proposed network (Appendix F, Section 5).

Table 4.3 A detailed comparison analysis and properties of the studied methods

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

4.2 A Robust Light-Weight Fused-Feature Encoder-Decoder Model for Monocular Facial Depth Estimation from Single Images Trained on Synthetic Data

This section provides an overview of the research paper: *Khan, Faisal, Waseem Shariff, Muhammad Ali Farooq, Shubhajit Basak, and Peter Corcoran. "A Robust Light-Weight Fused-Feature Encoder-Decoder Model for Monocular Facial Depth Estimation from Single Images Trained on Synthetic Data." Neural Networks (2022), [5], see Appendix G.*

Table 4.4 shows the author's contributions for the research works [5].

Table 4.4 Author's

Contributions to [5]

Contribution Criteria	Contribution Percent
Hypothesis/Idea for Research	FK 70%, MW 20%, MAF 10%
Experiments and Evaluations	FK 70%, MW 10%, MAF 10%, SB 10%
Background	FK 60%, MW 10%, MAF 10%, SB 10%, PC 10%
Preparation of the Manuscript	FK 60%, MW 10%, MAF 10%, SB 10%, PC 10%

4.2.1 Research Objectives

To further improve the accuracy of the SoA DL facial depth estimation methods, accurate datasets with pixel GT are required. These methods are suffering from low accuracy and large measurement noise [28]. Conventional systems utilize fully connected layers, which complicate the algorithms and needs additional memory, making them impractical for deployment on consumer devices and also suffer from issues like information loss that leads to low pixel values in depth images. The model applied in this research optimizes the acquisition of ideal parameters, hence minimizing model complexity during the facial depth estimation training procedure. The steps below are used to further explain the main work of the paper:

1. A lightweight neural facial depth estimation model based on single image frames is proposed.
2. By using a feature fusion module, the model employ pixel representations and recover full details in terms of facial features and boundaries.

3. It has a smaller number of parameters and is computationally much simpler.
4. An appropriate loss function is used that leads to higher performance.
5. The model performs better than existing comparative SoA facial depth networks in terms of its generalization ability and robustness across different test datasets, setting a new baseline method for facial depth maps.
6. The proposed model is converted to ONNX and it can be used for deployment in embedded systems and in Edge-AI applications. The rendered point clouds from a novel viewpoint is reconstructed and the results are compared.

This work consists of a basic encoder-decoder network design, the features are extracted by initializing the encoder with a high-performance pre-trained network and reconstructing high-quality facial depth maps with a simple decoder (Appendix G, Section 3). The model used pixel representations and recover full details in terms of facial features and boundaries by employing a feature fusion module. The model is composed of 22 layers, which are divided into eight parts: convolutional layers 1-5, a global average pooling (AP) layer, and a fully connected (FC) layer. The initial features are corrected in the channel dimension to increase the model intensity of learning features, allowing the model to recognize the key characteristics of various channels automatically. The global average pooling layer is then used in place of the fully connected layers to reduce model parameters, speed it up model convergence, and improve model accuracy. In the decoder stage, convolution is used to reduce the channel dimension of the bottleneck feature, thus further avoiding algorithm complexity. Then, to increase the size of the features, a series of bilinear upsampling layers are used. Finally, to estimate the facial depth map, two convolution layers and a sigmoid function are applied to the output. Furthermore, the depth map is scaled by the maximum depth value to give the depth in meters. In order to make better use of the precise details of the local structures, a skip connection is introduced into the proposed fusion module.

When tested and analyzed across four public facial depth datasets, the suggested network gives a more reliable SoA, with much less computational complexity and a reduced number of parameters (Appendix G, Section 4). The training technique is essentially based on the usage of synthetic human facial images, which provide a pixel accurate GT depth map, and the employment of an appropriate loss function leads to better performance (Appendix G, Section 5).

Our comprehensive experiments, which span roughly four GPU, demonstrate that a model trained on a rich and diverse set of images, when combined with an optimal training procedure generates SOA results in a range of situations Table 4.5. Table 4.5 shows the comparison

of various depth maps methods, BTS [90], Densedepth [91], UNet-simple [92], ResNet-101 [30], EfficientNet-B0 [93], MiDaS [94], DPT [95], LapDepth-Face [3], FaceDepth [30] on synthetic human facial depth dataset [30] with the proposed method LEDDEPTH which is briefly discussed in (Appendix G). The proposed network achieved 0.0203 RMSE as shown

Table 4.5 Comparison of various depth maps methods with the proposed method LEDDEPTH

No.	Methods	AbsRel	SqRel	RMSE	RMSElog	$\delta_1 < 1.25$	$\delta_2 < 1.25^2$	$\delta_3 < 1.25^3$
1.	DenseDepth-161	0.0296	0.0096	0.0373	0.0129	0.9890	0.9920	0.9981
2.	ResNet-101	0.0123	0.0210	0.0306	0.0089	0.9938	0.9960	0.9980
3.	BTS	0.0165	0.0092	0.0206	0.0102	0.9830	0.9943	0.9956
4.	EfficientNet-B0	0.0145	0.0280	0.0360	0.0154	0.9912	0.9934	0.9978
5.	UNet-simple	0.0103	0.0207	0.0281	0.0089	0.9960	0.9956	0.9987
6.	MiDaS	0.0146	0.0204	0.0356	0.0323	0.9665	0.9902	0.9983
7.	DPT	0.0156	0.0106	0.0394	0.0184	0.9567	0.9646	0.9943
8.	LapDepth-Face	0.0145	0.0041	0.0204	0.3614	0.9545	0.9857	0.9958
9.	FaceDepth	0.0176	0.0030	0.0205	0.1252	0.9642	0.9849	0.9951
10.	LEDDEPTH	0.0113	0.0025	0.0203	0.1172	0.9888	0.9961	0.9967

in Table. 4.2 (row 10) compared to the SOA methods.

4.2.2 Summary of Contributions

The study [4] offers a system for developing sophisticated synthetic human face datasets by incorporating various variations in synthetic facial data, a synthetic human model with a green3D model design, the iClone Character Creation workflow, and adding variations to models in iClone. This is followed by model transfer from iClone to Blender, model manipulation in Blender, creating 3D scenes in Blender, The Blender camera model, selecting 3D background environments in Blender, and GT rendering in Blender along with the dataset organization. The created synthetic human facial depth dataset is analyzed using SoA DESI neural networks. Additionally, a new CNN model is developed along with a hybrid loss function and its performance is compared to the SoA networks. Initially, SoA DESI techniques are trained on a synthetic human facial dataset, and their performance is compared to that of the proposed network.

The second paper improves on the previous paper results by using a ResNet in the encoder and a feature fusion module in the decoder stage, which makes the network structure better and simpler than the available SoA models. It also provides a comparison analysis of the 3D point cloud with the SOA and ONNX conversion method that can be used for consumer applications.

4.2.3 Discussion of Contributions

A critical result of this research work is that training neural facial depth networks on synthetic human facial data produces higher-quality depth maps. Using lightweight neural single image depth estimation models with the high-quality training data can estimate accurate facial depth maps. The performance of various methods is compared, including error and accuracy metrics with SOA on real datasets.

Deeper networks perform well, but their applications encounter a significant problem since estimation tasks take longer to perform when using deeper networks. The practical uses of depth estimation networks significantly impact their ability to function in real-time on embedded devices. In order to develop or design networks that can learn facial depth information, this chapter covers improved light-weight DL face depth estimation networks. These networks can have lower complexity and system memory needs, improved accuracy, and the ability to operate in real time on embedded devices.

Chapter 5

Additional Contributions

Several of my secondary publications are briefly discussed in this chapter. These papers mainly focus on learning 3D head positions from synthetic data and reconstructing 3D face models from a single camera frame. They are listed below:

- S. Basak, P. Corcoran, **Faisal K**, 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. It is available in Appendix H.
- Basak S, Javidnia H, **Khan F**, McDonnell R, Schukat M. Methodology for building synthetic datasets with virtual humans. In 2020 31st Irish Signals and Systems Conference (ISSC) 2020 Jun 11 (pp. 1-6). IEEE, which can be found in Appendix I.
- Basak S, **Khan F**, McDonnell R, Schukat M. Learning accurate head pose for consumer technology from 3D synthetic data. In 2021 IEEE International Conference on Consumer Electronics (ICCE) 2021 Jan 10 (pp. 1-6). IEEE, which can be found in Appendix J.

In [62], the process for constructing a synthetic head pose dataset using a commercially accessible 3D asset creation application, iClone, and an open-source 3D computer animation program, Blender. The experimental results indicated that training an SoA HPE model with the newly suggested dataset independently provides SoA HPE performances. By implementing the visual adversarial transfer learning method and training the model with labelled synthetic data and unlabeled real data, it is demonstrated that the model is capable of learning features that make and generate better results than training exclusively with synthetic data.

In [63] this paper, a methodology for synthetically generating facial data that can be utilized as a part of a toolset to build relatively big facial datasets with a high level of control

across facial and environmental variations is presented. These large datasets can be utilized to train deep neural networks more accurately and precisely. It utilizes a 3D morphable face model for the creation of numerous 2D images over a collection of 100 synthetic identities, offering complete control over image variations including poses, lighting, and environment.

In [96], a rendering process to build pixel-perfect synthetic 2D headshot images from high-quality 3D facial models with realistic pose angle labels is provided. A number of age, racial, and gender variations are supplied. More than 300k combinations of RGB images with their related head pose annotations are included in the generated collection. There are various changes in posture, lighting, and backgrounds for each one hundred 3D model. The data is analyzed by training and validating a SoA head pose estimation model against by the prominent benchmark dataset BIWI.

Chapter 6

Conclusions and Future Works

Various works completed over the course of this Ph.D. research have been introduced in previous chapters. We summarize the key findings of this research in the context of the main goals and objectives provided in the thesis introduction chapter 1. The following are the main contributions of this work.

- Single image depth maps and neural models
- Building 2D datasets from 3D models with pixel-accurate depth GT
- New models for facial depth estimation

The first and second chapters 1, 2 provide a full summary of the research work done, a summary of the contributions of the thesis, a list of publications and taxonomy including a problem description and a brief review of traditional depth estimation techniques. For monocular depth estimation, important datasets, loss functions, and SoA DL-based techniques are studied, analyzed, and discussed. We conclude the study by looking ahead to future research projects that will require more examination into monocular depth estimation difficulties, particularly for faces (Appendix A, B, C).

The third chapter 3 of this thesis provides a brief introduction to the large-scale facial 3D models created using synthetic data, and high-quality human facial depth generated from synthetic 3D models. The first objective of this study was to produce 3D virtual human models with corresponding RGB and depth images. The goal is to evaluate the accuracy of synthetic datasets used to address practical problems in various application fields. This should enable further methodological advancements to the datasets collected to address certain use case applications (Appendix D, E).

This thesis fourth chapter 4 introduces an enhanced light-weight neural networks for estimating facial depth maps. The SoA DESI neural networks are trained and tested using the generated synthetic human facial depth dataset.

New CNN models are proposed, and their performance is compared against SoA networks. The proposed models are more computationally efficient than the existing SoA depth estimation models and perform as well as or better than the SoA when evaluated across four public datasets. When tested and evaluated across public facial depth datasets, the suggested networks provide a more reliable SoA, with significantly less computational complexity and a reduced number of parameters. The training procedure is primarily based on the use of synthetic human facial images, which provide a consistent ground truth depth map, and the employment of appropriate loss functions leads to higher performance. Numerous experiments have been performed to validate and demonstrate the usefulness of the proposed approaches.

Finally, the models perform better than existing comparative facial depth networks in terms of generalization ability and robustness across different test datasets, setting new baseline methods for facial depth maps. Furthermore, the results obtained from the proposed models are used to reconstruct 3D rendering from a novel viewpoint. Point clouds rendered via Open3D are used. Also, the lightweight neural facial depth estimation model is converted to ONNX and it can be used for deployment in embedded systems and in Edge-AI applications. ONNX is a freely available format for encoding deep neural networks. With ONNX, application developers can more quickly integrate models between SoA packages and determine the ideal mix for their needs. A community of contributors contributes to the development and support of ONNX (Appendix F, G).

Dataset contributions: As described in chapter 3, the dataset contributions of this work consisted of novel synthetic facial depth data and data collection from 3D virtual human models. The generated synthetic dataset used in this research work consists of 100 3D virtual human models with approximately 3.5k 2D rendered RGB and GT depth images. The synthetic data is used to train a CNN-based facial depth estimation system, which is then validated on both synthetic and real-world images. 3D reconstruction, driver monitoring systems, robotic vision systems, and advanced scene understanding are all possible applications.

A fundamental conclusion of this research is that synthetic human facial data can provide greater quality ground truth depth data than real data. This high-quality training data can be used to create improved, lightweight single-image depth models which can provide depth information on lightweight computational devices.

Some of the main future directions include network optimisation. When facing high-quality images and aiming to forecast high-resolution depth, complex DL networks have high memory requirements, computational time, complex environments such as occlusions, highly cluttered scenes, and complex material properties of the scene and training datasets annotated with ground truth labels for depth estimation. However, devolving lighter DL-based architectures remains desirable especially if they are to be deployed in smart consumer devices.

Many different 3D reconstruction tasks using depth maps, such as segmentation and instance segmentation, can be addressed in future research. It can also be helpful in face identification and verification tasks to enhance the suggested 3D face models and the corresponding ground truth depth with additional variations, segmentation masks, and annotation with 2D-3D boxing.

Another future work looks into ways to superimpose highly detailed face textures over artificial avatar models and incorporate more complex facial dynamics, such as modifications in the mouth and eyes used to illustrate a variety of expressions.

An investigation of various lightweight encoder-decoder architectures focused on developing more robust neural networks, as well as paying more attention to the newly developed facial depth datasets to obtain pixel-accurate ground truth depth maps, data augmentation methods, and tests using a wider variety of test datasets would be of interest.

One of the most important future works will be to optimize all SoA methods for use on low-power computational systems. These technologies, particularly the algorithms provided in Tables 4.2 and 4.5, have a high potential for implementation on consumer devices.

Another noteworthy feature is that some of the proposed algorithms operate on high-resolution images without any downsampling, bringing them one step ahead of the state of the art. As previously stated, this field of study suffers from a lack of real-world data and appropriate evaluation metrics. This is another potential gap that should be addressed. Proposing a uniform assessment system and global measurements has the potential to significantly alter the way present approaches are evaluated.

References

- [1] 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, 2020.
- [2] F. Khan, S. Hussain, S. Basak, M. Moustafa, and P. Corcoran, “A review of benchmark datasets and training loss functions in neural depth estimation,” *IEEE Access*, vol. 9, pp. 148 479–148 503, 2021.
- [3] F. Khan, M. A. Farooq, W. Shariff, S. Basak, and P. Corcoran, “Towards monocular neural facial depth estimation: Past, present, and future,” *IEEE Access*, 2022.
- [4] F. , 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 Networks*, vol. 142, pp. 479–491, 2021.
- [5] F. Khan, W. Shariff, M. Farooq, S. Basak, and P. Corcoran, “A robust light-weight fused-feature encoder-decoder model for monocular facial depth estimation from single images trained on synthetic data,” *Available at SSRN 4217345*.
- [6] M. Alam, M. D. Samad, L. Vidyaratne, A. Glandon, and M. Iftekharuddin, “Survey on deep neural networks in speech and vision systems,” *Neurocomputing*, vol. 417, pp. 302–321, 2020.
- [7] W. Huang, J. Cheng, Y. Yang, and G. Guo, “An improved deep convolutional neural network with multi-scale information for bearing fault diagnosis,” *Neurocomputing*, vol. 359, pp. 77–92, 2019.
- [8] G. Tian, L. Liu, J. Ri, Y. Liu, and Y. Sun, “Objectfusion: An object detection and segmentation framework with rgb-d slam and convolutional neural networks,” *Neurocomputing*, vol. 345, pp. 3–14, 2019.
- [9] G. Kim, B. Park, and A. Kim, “1-day learning, 1-year localization: Long-term lidar localization using scan context image,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1948–1955, 2019.
- [10] T. Zhang, Y. Yang, Y. Zeng, and Y. Zhao, “Cognitive template-clustering improved linemod for efficient multi-object pose estimation,” *Cognitive Computation*, vol. 12, no. 4, pp. 834–843, 2020.
- [11] F. Mueller, M. Davis, F. Bernard, O. Sotnychenko, M. Verschoor, M. A. Otaduy, D. Casas, and C. Theobalt, “Real-time pose and shape reconstruction of two interacting hands with a single depth camera,” *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–13, 2019.

- [12] J. Zhang, Q. Su, C. Wang, and H. Gu, "Monocular 3d vehicle detection with multi-instance depth and geometry reasoning for autonomous driving," *Neurocomputing*, vol. 403, pp. 182–192, 2020.
- [13] H. Luo, Y. Gao, Y. Wu, C. Liao, X. Yang, and K.-T. Cheng, "Real-time dense monocular slam with online adapted depth prediction network," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 470–483, 2018.
- [14] J. Sun, Z. Wang, H. Yu, S. Zhang, J. Dong, and P. Gao, "Two-stage deep regression enhanced depth estimation from a single rgb image," *IEEE Transactions on Emerging Topics in Computing*, 2020.
- [15] Y.-M. Tsai, Y.-L. Chang, and L.-G. Chen, "Block-based vanishing line and vanishing point detection for 3d scene reconstruction," in *2006 international symposium on intelligent signal processing and communications*. IEEE, 2006, pp. 586–589.
- [16] C. Tang, C. Hou, and Z. Song, "Depth recovery and refinement from a single image using defocus cues," *Journal of Modern Optics*, vol. 62, no. 6, pp. 441–448, 2015.
- [17] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape-from-shading: a survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 21, no. 8, pp. 690–706, 1999.
- [18] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. IEEE, 1999, pp. 1150–1157.
- [19] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.
- [20] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *2007 IEEE 11th international conference on computer vision*. Ieee, 2007, pp. 1–8.
- [21] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.
- [22] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 25–39, 1983.
- [23] R. Xiong, S. Zhang, Z. Gan, Z. Qi, M. Liu, X. Xu, Q. Wang, J. Zhang, F. Li, and X. Chen, "A novel 3d-vision-based collaborative robot as a scope holding system for port surgery: a technical feasibility study," *Neurosurgical focus*, vol. 52, no. 1, p. E13, 2022.
- [24] M. Li, B. Huang, and G. Tian, "A comprehensive survey on 3d face recognition methods," *Engineering Applications of Artificial Intelligence*, vol. 110, p. 104669, 2022.
- [25] A. Mertan, D. J. Duff, and G. Unal, "Single image depth estimation: An overview," *Digital Signal Processing*, p. 103441, 2022.

- [26] V. G. V. A. Attention, “Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer.”
- [27] M. Song, S. Lim, and W. Kim, “Monocular depth estimation using laplacian pyramid-based depth residuals,” *IEEE transactions on circuits and systems for video technology*, vol. 31, no. 11, pp. 4381–4393, 2021.
- [28] D. Kim, W. Ga, P. Ahn, D. Joo, S. Chun, and J. Kim, “Global-local path networks for monocular depth estimation with vertical cutdepth,” *arXiv preprint arXiv:2201.07436*, 2022.
- [29] J. S. Katrolia, B. Mirbach, A. El-Sherif, H. Feld, J. Rambach, and D. Stricker, “Ticam: A time-of-flight in-car cabin monitoring dataset,” *arXiv preprint arXiv:2103.11719*, 2021.
- [30] 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 networks: the official journal of the International Neural Network Society*, vol. 142, pp. 479–491.
- [31] Y. Ming, X. Meng, C. Fan, and H. Yu, “Deep learning for monocular depth estimation: A review,” *Neurocomputing*, vol. 438, pp. 14–33, 2021.
- [32] A. N. Gorban, E. M. Mirkes, and I. Y. Tyukin, “How deep should be the depth of convolutional neural networks: a backyard dog case study,” *Cognitive Computation*, vol. 12, no. 2, pp. 388–397, 2020.
- [33] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz, “Neural rgb (r) d sensing: Depth and uncertainty from a video camera,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 986–10 995.
- [34] F. Liu, S. Zhou, Y. Wang, G. Hou, Z. Sun, and T. Tan, “Binocular light-field: Imaging theory and occlusion-robust depth perception application,” *IEEE Transactions on Image Processing*, vol. 29, pp. 1628–1640, 2019.
- [35] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [36] H. Javidnia and P. Corcoran, “A depth map post-processing approach based on adaptive random walk with restart,” *IEEE Access*, vol. 4, pp. 5509–5519, 2016.
- [37] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [39] 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*, 2016, pp. 770–778.

- [40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
- [41] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, “Ghostnet: More features from cheap operations,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1580–1589.
- [42] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.
- [43] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.
- [44] S. Ji, W. Xu, M. Yang, and K. Yu, “3d convolutional neural networks for human action recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2012.
- [45] L. Wang, W. Li, W. Li, and L. Van Gool, “Appearance-and-relation networks for video classification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1430–1439.
- [46] K. Gwn Lore, K. Reddy, M. Giering, and E. A. Bernal, “Generative adversarial networks for depth map estimation from rgb video,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 1177–1185.
- [47] H. Jung, Y. Kim, D. Min, C. Oh, and K. Sohn, “Depth prediction from a single image with conditional adversarial networks,” in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 1717–1721.
- [48] H. Chen, Y. Yan, J. Qin, T. Zhao, and T. Guo, “Recognition-oriented facial depth estimation from a single image,” *Applied Intelligence*, pp. 1–19, 2022.
- [49] Z. Zhu, A. Su, H. Liu, Y. Shang, and Q. Yu, “Vision navigation for aircrafts based on 3d reconstruction from real-time image sequences,” *Science China Technological Sciences*, vol. 58, no. 7, pp. 1196–1208, 2015.
- [50] X. Chai, F. Gao, C. Qi, Y. Pan, Y. Xu, and Y. Zhao, “Obstacle avoidance for a hexapod robot in unknown environment,” *Science China Technological Sciences*, vol. 60, no. 6, pp. 818–831, 2017.
- [51] S.-J. Park, K.-S. Hong, and S. Lee, “Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4980–4989.
- [52] L. Zou and Y. Li, “A method of stereo vision matching based on opencv,” in *2010 International conference on audio, language and image processing*. IEEE, 2010, pp. 185–190.

- [53] L. R. Ramírez-Hernández, J. C. Rodríguez-Quinoñez, M. J. Castro-Toscano, D. Hernández-Balbuena, W. Flores-Fuentes, R. Rascón-Carmona, L. Lindner, and O. Sergiyenko, "Improve three-dimensional point localization accuracy in stereo vision systems using a novel camera calibration method," *International Journal of Advanced Robotic Systems*, vol. 17, no. 1, p. 1729881419896717, 2020.
- [54] F. Mancini, M. Dubbini, M. Gattelli, F. Stecchi, S. Fabbri, and G. Gabbianelli, "Using unmanned aerial vehicles (uav) for high-resolution reconstruction of topography: The structure from motion approach on coastal environments," *Remote sensing*, vol. 5, no. 12, pp. 6880–6898, 2013.
- [55] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6243–6252.
- [56] K. Yoneda, H. Tehrani, T. Ogawa, N. Hukuyama, and S. Mita, "Lidar scan feature for localization with highly precise 3-d map," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 1345–1350.
- [57] G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara, "Poseidon: Face-from-depth for driver pose estimation," 2017, pp. 4661–4670.
- [58] R. Min, N. Kose, and J.-L. Dugelay, "Kinectfacedb: A kinect database for face recognition," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 11, pp. 1534–1548, 2014.
- [59] G. Fanelli, T. Weise, J. Gall, and L. Van Gool, "Real time head pose estimation from consumer depth cameras," *Joint pattern recognition symposium*, pp. 101–110, 2011.
- [60] F. , S. Basak, H. Javidnia, M. Schukat, and P. Corcoran, "High-accuracy facial depth models derived from 3d synthetic data," in *2020 31st Irish Signals and Systems Conference (ISSC)*. IEEE, 2020, pp. 1–5.
- [61] F. , S. Basak, and P. Corcoran, "Accurate 2d facial depth models derived from a 3d synthetic dataset," in *2021 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2021, pp. 1–6.
- [62] S. Basak, P. Corcoran, F. , R. McDonnell, and M. Schukat, "Learning 3d head pose from synthetic data: A semi-supervised approach," *IEEE Access*, vol. 9, pp. 37 557–37 573, 2021.
- [63] S. Basak, H. Javidnia, F. , R. McDonnell, and M. Schukat, "Methodology for building synthetic datasets with virtual humans," in *2020 31st Irish Signals and Systems Conference (ISSC)*. IEEE, 2020, pp. 1–6.
- [64] L. Chen, W. Tang, N. W. John, T. R. Wan, and J. J. Zhang, "Augmented reality for depth cues in monocular minimally invasive surgery," *arXiv preprint arXiv:1703.01243*, 2017.
- [65] X. Liu, A. Sinha, M. Ishii, G. D. Hager, A. Reiter, R. H. Taylor, and M. Unberath, "Dense depth estimation in monocular endoscopy with self-supervised learning methods," *IEEE transactions on medical imaging*, vol. 39, no. 5, pp. 1438–1447, 2019.

- [66] P. R. Palafox, J. Betz, F. Nobis, K. Riedl, and M. Lienkamp, "Semanticdepth: Fusing semantic segmentation and monocular depth estimation for enabling autonomous driving in roads without lane lines," *Sensors*, vol. 19, no. 14, p. 3224, 2019.
- [67] T. Laidlow, J. Czarowski, and S. Leutenegger, "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)*. IEEE, 2019, pp. 4068–4074.
- [68] Y. Dai, H. Li, and M. He, "Projective multiview structure and motion from element-wise factorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 9, pp. 2238–2251, 2013.
- [69] H. Javidnia and P. Corcoran, "Accurate depth map estimation from small motions," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2453–2461.
- [70] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 270–279.
- [71] Y. Kuznetsov, J. Stuckler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6647–6655.
- [72] S. Bazrafkan, H. Javidnia, J. Lemley, and P. Corcoran, "Semiparallel deep neural network hybrid architecture: first application on depth from monocular camera," *Journal of Electronic Imaging*, vol. 27, no. 4, p. 043041, 2018.
- [73] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2002–2011.
- [74] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, "Structured attention guided convolutional neural fields for monocular depth estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3917–3925.
- [75] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR 2011*. IEEE, 2011, pp. 1521–1528.
- [76] C. Wang, S. Lucey, F. Perazzi, and O. Wang, "Web stereo video supervision for depth prediction from dynamic scenes," in *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 348–357.
- [77] W. Chen, Z. Fu, D. Yang, and J. Deng, "Single-image depth perception in the wild," *Advances in neural information processing systems*, vol. 29, pp. 730–738, 2016.
- [78] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3260–3269.

- [79] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, “Tanks and temples: Benchmarking large-scale scene reconstruction,” *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.
- [80] “3d animation software: iclone: Reallusion. (n.d).”
- [81] “Foundation, b. (n.d.). home of the blender project - free and open 3d creation software.”
- [82] Y. Roh, G. Heo, and S. E. Whang, “A survey on data collection for machine learning: a big data-ai integration perspective,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1328–1347, 2019.
- [83] J. Ren, A. Hussain, J. Zheng, C.-L. Liu, and B. Luo, “Special issue on recent advances in cognitive learning and data analysis,” *Cognitive Computation*, vol. 13, no. 4, pp. 785–786, 2021.
- [84] P. Zama Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Di Stefano, “Geometry meets semantics for semi-supervised monocular depth estimation,” in *Asian Conference on Computer Vision*. Springer, 2018, pp. 298–313.
- [85] E. D. P. Supervisor, “A preliminary opinion on data protection and scientific research,” 2020.
- [86] S. Straková, “Human-computer interaction in the context of gdpr: How web users perceive and respond to blocking vs. non-blocking privacy pop-up notices,” Ph.D. dissertation, Tilburg University, 2021.
- [87] 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.
- [88] R. Min, N. Kose, and J.-L. Dugelay, “Kinectfacedb: A kinect database for face recognition,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 11, pp. 1534–1548, 2014.
- [89] G. Fanelli, T. Weise, J. Gall, and L. V. Gool, “Real time head pose estimation from consumer depth cameras,” in *Joint pattern recognition symposium*. Springer, 2011, pp. 101–110.
- [90] 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.
- [91] I. Alhashim and P. Wonka, “High quality monocular depth estimation via transfer learning,” *arXiv preprint arXiv:1812.11941*, 2018.
- [92] F. Khan, S. Basak, and P. Corcoran, “Accurate 2d facial depth models derived from a 3d synthetic dataset,” in *2021 IEEE International Conference on Consumer Electronics (ICCE)*, 2021, pp. 1–6.
- [93] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

-
- [94] 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 transactions on pattern analysis and machine intelligence*, 2020.
- [95] R. Ranftl, A. Bochkovskiy, and V. Koltun, “Vision transformers for dense prediction,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 179–12 188.
- [96] S. Basak, F. , R. McDonnell, and M. Schukat, “Learning accurate head pose for consumer technology from 3d synthetic data,” in *2021 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2021, pp. 1–6.

Appendix A

Deep Learning-Based Monocular Depth Estimation Methods—A State-of-the-Art Review

Review

Deep Learning-Based Monocular Depth Estimation Methods—A State-of-the-Art Review

Faisal Khan ¹, Saqib Salahuddin ¹ and Hossein Javidnia ^{2,*} 

¹ College of Engineering and Informatics, National University Ireland Galway, Galway H91 TK33, Ireland; f.khan4@nuigalway.ie (F.K.); saqib.salahuddin@nuigalway.ie (S.S.)

² ADAPT Centre, Trinity College Dublin, Dublin D02 PN40, Ireland

* Correspondence: hossein.javidnia@tcd.ie

Received: 27 February 2020; Accepted: 12 April 2020; Published: 16 April 2020



Abstract: Monocular depth estimation from Red-Green-Blue (RGB) images is a well-studied ill-posed problem in computer vision which has been investigated intensively over the past decade using Deep Learning (DL) approaches. The recent approaches for monocular depth estimation mostly rely on Convolutional Neural Networks (CNN). Estimating depth from two-dimensional images plays an important role in various applications including scene reconstruction, 3D object-detection, robotics and autonomous driving. This survey provides a comprehensive overview of this research topic including the problem representation and a short description of traditional methods for depth estimation. Relevant datasets and 13 state-of-the-art deep learning-based approaches for monocular depth estimation are reviewed, evaluated and discussed. We conclude this paper with a perspective towards future research work requiring further investigation in monocular depth estimation challenges.

Keywords: monocular depth estimation; single image depth estimation; CNN monocular depth

1. Introduction

Monocular depth estimation is a fundamental challenge in computer vision and has potential applications in robotics, scene understanding, 3D reconstruction and medical imaging [1–4]. This problem remains challenging as there are no reliable cues for perceiving depth from a single image. For example, temporal information and stereo correspondences are missing from such images. The classical depth estimation approaches heavily rely on multi-view geometry [5–9] such as stereo image [10,11]. These methods require alignment and calibration procedures which are important for multi-camera or multi-sensor depth measurement systems [12,13]. Multi-view methods acquire depth information by utilising visual cues and different camera parameters.

Most of the binocular or multi-view methods are able to estimate fairly accurate depth information. However, their computational time and memory requirements are important challenges for many applications [14]. The idea of using the monocular image to capture depth information could potentially solve the memory requirement issue, but it is computationally difficult to capture the global properties of a scene such as texture variation or defocus information.

Recently, the advancement of Convolutional Neural Networks (CNN) and publicly available datasets have significantly improved the performance of monocular depth estimation methods [15–19].

This paper offers a comprehensive and structured survey of deep learning-based monocular depth estimation approaches. The goal of the review is to assist the reader to navigate this emerging field, which has become of significant interest to the computer vision community in recent years. The rest of the survey is organized as follows: Section 2 presents a summary and basic concept of monocular depth estimation, problem description, traditional methods for depth estimation and publicly available datasets. Section 3 reviews the recent deep learning architectures for monocular depth estimation

categorised in supervised, self-supervised and semi-supervised methods. Section 4 compares the state-of-the-art approaches followed by discussion and potential future research directions presented in Section 5.

2. An Overview of Monocular Depth Estimation

The concept of depth estimation refers to the process of preserving 3D information of the scene using 2D information captured by cameras. Monocular solutions tend to achieve this goal using only one image. These methods aim to estimate distances between scene objects and the camera from one viewpoint. This requires the method to perform depth estimation on low-cost embedded systems. There are a variety of devices commercially available to provide depth information, however, their processing power, computational time, range limitation and cost make them impractical for consumer devices. Sensors such as Kinect are commonly used in consumer devices [20,21]. These types of sensor are categorized as Time-of-Flight (ToF) where the depth information is acquired by calculating the time required for a ray of light to travel from a light source to an object and back to the sensor [22]. ToF sensors are more suitable for the indoor environment and short range (<2 m) depth sensing. On the other hand, laser-based scanners (LiDAR) are commonly utilised for 3D measurement in the outdoor environment. The key advantages of LiDAR sensors are high resolution, accuracy, performance in low light and speed. However, LiDARs are expensive devices and they require extensive power resources which make them unsuitable for consumer products.

It has been shown in the state-of-the-art that monocular depth estimation methods could be a potential solution to address many of these challenges [23–25]. These methods perform with a relatively small number of operations and in less computation time. They do not require alignment and calibration which is important for multi-camera, or multi-sensor depth measurement systems. Accurate monocular depth estimation methods can play an important role in understanding 3D scene geometry and 3D reconstruction, particularly in cost-sensitive applications and use cases.

2.1. Problem Representation

Let $I \in \mathbb{R}^{w \times h}$ be an image with size $w \times h$. The goal is to estimate the corresponding depth information $D \in \mathbb{R}^{w \times h}$. This is an ill-posed problem as there is an ambiguity in the scale of the depth. Supervised learning-based methods try to address this issue by approximately learning the scale from a set of training images. On the other hand, unsupervised and semi-supervised methods often utilise an extra input for training such as stereo image sets, visual odometry and 6D camera pose estimation to tackle the scale ambiguity issue. These methods mathematically define the problem as follows: given a large dataset of Red-Green-Blue (RGB) and depth images, single image depth estimation can be considered as a regression problem that uses a standard loss function such as Mean Square Error (MSE). To achieve this, a training set τ can be represented as follows:

$$\tau = \{(I_n, D_n)\}, I_n \in \mathbb{R}^{w \times h} \quad \text{and} \quad D_n \in \mathbb{R}^{w \times h} \quad (1)$$

2.2. Traditional Methods for Depth Estimation

Most of the traditional methods for depth estimation rely on the assumption of having observations of the scene, either in space or time (e.g., stereo or multi-view, structure from motion) [10,11,26,27]. Traditional methods can be categorized in two sets, active and passive methods.

Active methods involve computing the depth in the scene by interacting with the objects and the environment. There are different types of active method, such as light-based depth estimation, which uses the active light illumination to estimate the distance to different objects. Ultrasound and ToF are other examples of active methods. These methods use the known speed of the wave to measure the time an emitted pulse takes to arrive at an image sensor. Passive methods exploit the optical features of captured images. These methods involve extracting the depth information by computational image

processing. In the category of passive methods, there are two primary approaches: (a) multi-view depth estimation, such as depth from stereo, and (b) monocular depth estimation.

The traditional depth estimation methods are mainly focused on multi-view geometry. The detailed review of those methods is outside the scope of this work. However, it is worth noting that multi-view traditional methods have various limitations including computational complexity and associated high energy requirements. Current research works take advantage of deep-learning methods to achieve more accurate results with lower computational and energy demands [15–19]. Deep learning-based approaches and the availability of large-scale datasets have significantly transformed the monocular depth estimation methods.

2.3. Datasets for Depth Estimation

A number of important datasets are particularly preferred for the depth estimation problem as they provide images and corresponding depth maps from different viewpoints. The following section highlights the popular datasets used to analyse the scenes. Consumer-level sensors such as the Kinect and Velodyne laser scanner [20,21,28] are commonly used to capture the ground truth depth images for datasets. A summary is presented in Table 1.

NYU-v2: the NYU-v2 dataset for depth estimation was introduced in [29]. The dataset consists of 1449 RGB images densely labelled with depth images. The datasets consist of 407K frames of 464 scenes taken from three different cities. These datasets are used for indoor scenes depth estimation, segmentation and classification.

Make3D: the Make3D dataset, introduced in [30], contains 400 and 134 outdoor images for training and testing, respectively. This dataset contains different types of outdoor, indoor and synthetic scenes that are used for depth estimation by presenting a more complex set of features.

KITTI: the KITTI dataset, introduced in [31], has two versions and is made of 394 road scenes providing RGB stereo sets and corresponding ground truth depth maps. The KITTI dataset is further divided into RD: KITTI Raw Depth [31]; CD: KITTI Continuous Depth [31,32]; SD: KITTI Semi-Dense Depth [31,32]; ES: Eigen Split [33]; ID: KITTI Improved Depth [34]. KITTI datasets are commonly used for different tasks including 3D object detection and depth estimation. The high-quality ground truth images are captured using the Velodyne laser scanner.

Pandora: the Pandora dataset, introduced [35], contains 250K full resolution RGB and corresponding depth images having their corresponding annotation. Pandora dataset is used for head centre localization, head pose estimation and shoulder pose estimation.

SceneFlow: this was introduced in [36] as one of the very first large-scale synthetic datasets consist of 39K stereo images with corresponding disparity, depth, optical flow and segmentation masks.

Table 1. Datasets for monocular depth estimation.

Dataset	Labelled Images	Annotation	Brief Description
NYU-v2 [29]	1449	Depth + Segmentation	Red-green-blue (RGB) and depth images taken from indoor scenes.
Make3D [30]	534	Depth	RGB and depth images taken from outdoor scenes.
KITTI [31]	94K	Depth aligned with RAW data + Optical Flow	RGB and depth from 394 road scenes.
Pandora [35]	250K	Depth + Annotation	RGB and depth images.
SceneFlow [36]	39K	Depth + Disparity + Optical Flow + Segmentation Map	Stereo image sets rendered from synthetic data with ground truth depth, disparity and optical flow.

3. Deep Learning and Monocular Depth Estimation

There has been a significant improvement in learning-based monocular depth estimation methods over the past couple of years [37–42]. The majority 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 self-supervised. Supervised methods accept a single image and the corresponding depth information for training. In such a case, the trained network can directly output the depth information. However, a large amount of high-quality depth data is required, which is hard to generalize to all use cases.

To overcome the need for high-quality depth estimation as seed data, numerous semi-supervised methods are proposed. Semi-supervised approaches require smaller amount of labelled data and a large amount of unlabeled data for training [16,43,44]. The limitation of semi-supervised methods is that the networks are unable to correct their own bias and require additional domain information such as camera focal length and sensor data.

Self-supervised methods only require a small number of unlabeled images to train the networks for depth estimation [15,42,45]. These methods obtain the depth information automatically by relating different input modalities. Self-supervised methods suffer from generalization issues. The models can only perform on a very limited set of scenarios with similar distribution as the training set.

Table 2 categorizes thirteen methods reviewed comprehensively in the next sub-sections into supervised, semi-supervised and self-supervised.

Table 2. Categories of deep learning-based monocular depth estimation methods (FC: fully convolutional; CNN: convolutional neural networks).

Method	Architecture	Category
EMDEOM [32]	FC	Supervised
ACAN [46]	Encoder-Decoder	
DenseDepth [47]	Encoder-Decoder	
DORN [18]	CNN	
VNL [48]	Encoder-Decoder	
BTS [49]	Encoder-Decoder	
DeepV2D [50]	CNN	
LISM [51]	Encoder-Decoder	Self-supervised
monoResMatch [38]	CNN	
PackNet-SfM [52]	CNN	
VOMonodepth [53]	Auto-Decoder	
monodepth2 [42]	CNN	
GASDA [54]	CNN	Semi-supervised

3.1. Supervised Methods

Rosa et al. [32] proposed a supervised framework to estimate continuous depth maps from LiDAR points. The framework utilises Hilbert Maps methodology [55] to generate dense depth map from the sparse point cloud projected from LiDAR scanner. Furthermore, the proposed framework takes advantage of the Fully Convolutional Residual Network (FCRN) proposed by Laina et al. [56] for depth estimation. The network is trained on the densified depth images which are augmented by flipping and applying colour distortion. Despite the comparable performance of this method against the state-of-the-art methods, it can only produce depth maps with 128×160 pixel resolution. More importantly, the network is biased by the output of the Hilbert maps' densification process which does not represent the truth depth information of the missing areas.

Yuru et al. [46] proposed a new supervised algorithm called the Attention-Based Context Aggregation Network (ACAN) to estimate depth maps. The algorithm utilises the deep residual architecture [57], dilated layer and self-attention module [58–60] to control the spatial scale and continuous pixel-level dense depth estimation. Moreover, the self-attention module creates a relationship among every pixel resulting in learning the attention weights and contextual information which can produce more accurate depth information. Furthermore, the algorithm uses image-pooling to combine the image-level information for depth estimation. Soft-ordinal inference translation is

used to transform the predicted probabilities into continuous depth values to produce more realistic depth maps. The network is trained on resized and cropped images from NYU-v2 [29] and KITTI [31] datasets. The context adaption feature of this network results in sharp boundaries in the structure of the predicted depth map.

Ibraheem et al. [47] proposed a supervised method to estimate depth maps with the help of transfer learning. The method utilises a CNN for estimating high-quality depth maps. The method uses standard encoder-decoder network architecture based on pre-trained DenseNet-169 [61] and ImageNet [62] networks for features extraction. Furthermore, the information obtained is passed to the decoder to calculate the final depth maps with the sampling layer [63]. The network is trained on the densified depth images, which are augmented by horizontal flipping and applying the colour distortion including swapping the green and red channels of the input images. It produces depth maps with 320×240 pixel resolution and is likely to be biased by the output of the bilinear upsampling layer which does not represent the accurate depth information for all regions.

Fu et al. [18] proposed a supervised method to estimate depth maps from the Spacing-Increasing Discretization (SID) approach. The framework utilises the dense feature extractor, cross channel information learner, multi-scale feature learner, encoder and ordinal regression optimizer for high-quality depth estimation. Furthermore, the network is defined in a simpler way that avoids needless subsampling and captures multi-scale information to save computational cost and time. The subsampling layers are removed in the pooling layers and dilated convolutions are added to obtain more accurate depth information. The network is trained on four challenging datasets including Make3D [30], NYU-v2 [29], KITTI [31] and ScanNet [64] to introduce more feature variations.

Yin et al. [48] proposed a supervised framework to estimate depth maps by taking advantage of the 3D geometric constraints. A simple type of geometric constraints known as ‘virtual norm’ is implemented which is determined by randomly sampled three points in the 3D reconstruction to obtain a high-quality depth estimation. Further, the method can estimate 3D structures of the scene and surface normals directly from depth maps.

The method uses the 3D geometric constraints to convert the estimated depth to 3D point cloud representations. The network is trained on the densified depth images which are augmented by randomly cropping and flipping. This method can produce depth maps with 384×512 pixel resolution which are more robust and have strong global constraints.

Jin et al. [49] proposed a supervised method for monocular depth estimation that uses new Local Planar Guidance Layers (LPGL) inserted into the decoding phase of the network. The method utilises a decoding stage with spatial resolutions of 1/8, 1/4 and 1/2 by placing a layer that guides the input features to the desired depth. Furthermore, a Dense Feature Extractor (DFE), Contextual Information Extractor (CIE), LPGL and their dense features are used for final depth estimation. The proposed framework takes advantage of the dense Atrous Apatial Pyramid Pooling layer [65] for depth estimation. The network is trained on random crop of size 352×704 for KITTI [31] and 416×544 for NYU-v2 [29] datasets.

Zachary et al. [50] targeted the issues of monocular depth estimation in videos. The proposed method known as DeepV2D combines two classical algorithms in an end-to-end architecture. The network consists of two modules, depth estimation and camera motion. The depth module takes the camera motion as input and returns an initial depth map. The camera motion module takes the predicted depth and outputs the refined camera motion. Furthermore, the network alternates between these two modules to predict the final depth map. The network is trained on four challenging datasets including Make3D [30], NYU-v2 [29], KITTI [31] and ScanNet [64] to introduce more feature variations and high quality depth estimation.

3.2. Self-Supervised Methods

Matan et al. [51] proposed a self-supervised method to estimate depth maps from Siamese networks [66] approaches. The method utilises the Siamese DispNet [36], ResNet [57] and VGG [67]

based network architectures for depth estimation. Further, the method predicts multi-scale disparity maps in four scales which are later concatenated with previous decoder layer output and the corresponding encoder output using the skip connections. The network is trained on the RGB and ground truth depth images with 1242×375 pixel resolution. The proposed network has the advantage of sharing weights to reduce computational operations by cutting the network size to half which could lead to a potential model for consumer devices.

Aleotti et al. [38] proposed a self-supervised framework to estimate depth maps using end-to-end monocular residual matching known as monoResMatch. The framework utilises stereo matching approach for depth estimation. The RGB image is mapped to the feature space and then synthesized to obtain features aligned with virtual right images. The network further considers high dimensional features at input image resolution to find multi-scale inverse depth map aligned with the input image. The model is constructed based on an hourglass structure with skip connections. The final stage consists of a disparity refinement module which estimates residual corrections to the initial disparity. The network is trained using Structural Similarity (SSIM) reconstruction loss, disparity smoothness loss with an edge-aware term and reverse Huber loss [68]. The model is trained on Cityscape [69] and KITTI [31] datasets with random crops of size 640×192 .

Guizilini et al. [52] proposed a self-supervised method to estimate depth maps by combining the geometry of the PackNet. The method utilises the symmetrical packing and unpacking blocks to combine the encoded and decoded information using 3D convolutions. The network follows a similar architecture as [70], which provides the encoder-decoder layers with skip connections having geometrical information of the dense depth estimation. Furthermore, the method introduces new packing and unpacking blocks having visual information for fine-grained high-resolution depth predictions. This model is trained on the RGB and ground truth depth images with 640×192 pixel resolution from unlabelled data which can be generalized into unseen environments. The proposed architecture uses upsampling and downsampling operations which increase the number of the parameters and result in inaccurately scaled depth maps.

Andraghetti et al. [53] employed a state-of-the-art visual odometry method to obtain 3D points and sparse depth maps. Furthermore, the sparse data is fed to a sparse auto-encoder to obtain a denser depth map. The output of this stage along with the corresponding RGB image are fed to a CNN to acquire a final densified depth map in a self-supervised manner. The network is trained on the RGB and ground truth depth images from the KITTI [31] dataset and predicts depth maps with 256×512 pixel resolution.

Clement et al. [42] proposed a self-supervised approach to estimate depth maps utilising a combination of three architectures and loss functions. The pipeline takes advantage of a fully connected U-Net [71] to predict depth and a pose network to estimate the pose between pairs of images. ResNet-18 [57] is selected as the encoder and the pre-trained ImageNet [62] model is used to initialise the weights. The proposed framework utilises appearance-based loss and it introduces a modified per-pixel minimum reprojection loss. The network is trained on KITTI [31] dataset with Eigen split and it estimate depth maps with 640×192 pixel resolution.

3.3. Semi-Supervised Methods

Shanshan et al. [54] proposed GASDA, a semi-supervised method to estimate depth maps using the geometry-aware symmetric domain adaption. This approach targets the generalisation issue of the depth estimation methods by training the model on synthetic data to estimate depth from natural images. The method uses symmetric style image translation and monocular depth prediction. Utilising the CycleGAN [72], GASDA involves both real to unreal and unreal to real image translations together with an epipolar geometry of the real stereo images. The network is trained with two image style translations and symmetric depth estimators to produce depth maps with 192×640 pixel resolution.

4. Evaluation Matrices and Criteria

The most commonly used quantitative metrics for evaluating the performance of monocular depth estimation methods are Absolute Relative Difference (AbsRel), Root Mean Square Error (RMSE), RMSE (log) and Square Relative Error (SqRel).

These metrics are defined as follows:

$$\text{AbsRel} = \frac{1}{N} \sum \frac{|d_i - d_i^*|}{d_i} \quad (2)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum |d_i - d_i^*|^2} \quad (3)$$

$$\text{RMSE}(\log) = \sqrt{\frac{1}{N} \sum |\log d_i - \log d_i^*|^2} \quad (4)$$

$$\text{SqRel} = \frac{1}{N} \sum \frac{|d_i - d_i^*|^2}{d_i} \quad (5)$$

$$\text{Accuracy with threshold } (\delta < thr) : \% \text{ of } d_i \text{ such that } \max\left(\frac{d_i}{d_i^*}, \frac{d_i^*}{d_i}\right) < thr, \quad (6)$$

where $thr = 1.25, 1.25^2, 1.25^3$

where d_i and d_i^* are the ground truth and predicted depth at pixel i and N is the total number of pixels.

All of the methods described in this section are tested on either KITTI [31] or NYU-v2 [29] datasets. In order to evaluate and compare all the methods, we used the publicly available pre-trained models. The main advantage of comparing the pre-trained models on both datasets is that it allows us to measure the generalised performance of the networks on different test sets. Table 3 illustrates the properties of the networks studied for monocular depth estimation including their input/output dimensions, number of parameters, Graphical Processing Unit (GPU) specification and the type of the architecture employed.

Table 3. Properties of the studied methods for monocular depth estimation (FC: fully convolutional; ED: encoder-decoder; AD: auto-decoder; CNN: convolutional neural networks; K: trained on KITTI; N: trained on NYU-v2).

Method	Input	Type	Optimizer	Parameters	Output	GPU Memory	GPU Model
BTS [49]	352 × 704 K	ED	Adam	47M	352 × 704 K	4 × 11 GB	1080 Ti
DORN [18]	385 × 513 K	CNN	Adam	123.4M	513 × 385 K	12 GB	TITAN Xp
VNL [48]	384 × 384 N	ED	SGD	2.7M	384 × 384 N	N/A	N/A
ACAN [46]	256 × 352 N	ED	SGD	80M	256 × 352 N	11 GB	1080 Ti
VOMonodepth [53]	256 × 512 K	AD	Adam	35M	256 × 512 K	12 GB	TITAN Xp
LSIM [51]	1242 × 375 K	ED	Adam	73.3M	1242 × 375 K	12 GB	TITAN Xp
GASDA [54]	192 × 640 K	CNN	Adam	70M	192 × 640 K	N/A	N/A
DenseDepth [47]	640 × 480 N	ED	Adam	42.6M	320 × 240 N	4 × 12 GB	TITAN Xp
monoResMatch [38]	192 × 640 K	CNN	Adam	42.5M	192 × 640 K	12 GB	TITAN Xp
EMDEOM [32]	304 × 228 K	FC	Adam	63M	128 × 160 K	12 GB	TITAN Xp
PackNet-SfM [52]	640 × 192 K	CNN	Adam	128M	640 × 192 K	8 × 16 GB	Tesla V100
monodepth2 [42]	640 × 192 K	CNN	Adam	70M	640 × 192 K	12 GB	TITAN Xp
DeepV2D [50]	640 × 480 N	CNN	RMSProp	32M	640 × 480 N	11 GB	1080 Ti

Table 4 presents the performance evaluation of the studied methods on KITTI [31] dataset. All the numbers presented in this table are reported by the respective authors. As shown in Table 4, DeepV2D [50] marginally achieved the best accuracy on the KITTI [31] dataset. The last four columns in this table represent the evaluation using RMSE (log) metric and threshold inlier measures defined in Equation (6). Not all the methods in Table 4 are trained and evaluated on the same part of the KITTI [31]

dataset. The Train and Test columns in Table 4 indicate the subsets of the KITTI [31] dataset used by each method.

Table 4. Evaluation results on KITTI dataset. Best method per metric is emboldened and highlighted in green. (RD: KITTI Raw Depth [31]; CD: KITTI Continuous Depth [31,32]; SD: KITTI Semi-Dense Depth [31,32]; ES: Eigen Split [33]; ID: KITTI Improved Depth [34]).

Method	Train	Test	Abs Rel	Sq Rel	RMSE	RMSElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
BTS [49]	ES(RD)	ES(RD)	0.060	0.182	2.005	0.092	0.959	0.994	0.999
DORN [18]	ES(RD)	ES(RD)	0.071	0.268	2.271	0.116	0.936	0.985	0.995
VNL [48]	ES(RD)	ES(RD)	0.072	0.883	3.258	0.117	0.938	0.990	0.998
ACAN [46]	ES(RD)	ES(RD)	0.083	0.437	3.599	0.127	0.919	0.982	0.995
VOMonodepth [53]	ES(RD)	ES(RD)	0.091	0.548	3.790	0.181	0.892	0.956	0.979
LSIM [51]	FT	RD	0.169	0.6531	3.790	0.195	0.867	0.954	0.979
GASDA [54]	ES(RD)	ES(RD)	0.143	0.756	3.846	0.217	0.836	0.946	0.976
DenseDepth [47]	ES(RD)	ES(RD)	0.093	0.589	4.170	0.171	0.886	0.965	0.986
monoResMatch [38]	ES(RD)	ES(RD)	0.096	0.673	4.351	0.184	0.890	0.961	0.981
EMDEOM [32]	RD, CD	SD	0.118	0.630	4.520	0.209	0.898	0.966	0.985
monodepth2 [42]	ES(RD)	ES(RD)	0.115	0.903	4.863	0.193	0.877	0.959	0.981
PackNet-SfM [52]	ES(RD)	ID	0.078	0.420	3.485	0.121	0.931	0.986	0.996
DeepV2D [50]	ES(RD)	ES(RD)	0.037	0.174	2.005	0.074	0.977	0.993	0.997

In another evaluation on the NYU-v2 [29] dataset, as shown in Table 5, DeepV2D [50] marginally achieved the best accuracy with very close performance to BTS [49]. The significant advantage of this method against the state-of-the-art is a learnable approach for a geometrical principal of structure from motion and relative camera pose estimation.

Table 5. Evaluation results on NYU-v2 dataset. Best method per metric is emboldened and highlighted in green.

Method	Abs Rel	Sq Rel	RMSE	RMSElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
BTS [49]	0.112	0.025	0.352	0.047	0.882	0.979	0.995
VNL [48]	0.113	0.034	0.364	0.054	0.815	0.990	0.993
DenseDepth [47]	0.123	0.045	0.465	0.053	0.846	0.970	0.994
ACAN [46]	0.123	0.101	0.496	0.174	0.826	0.974	0.990
DORN [18]	0.138	0.051	0.509	0.653	0.825	0.964	0.992
monoResMatch [38]	1.356	1.156	0.694	1.125	0.825	0.965	0.967
monodepth2 [42]	2.344	1.365	0.734	1.134	0.826	0.958	0.979
EMDEOM [32]	2.035	1.630	0.620	1.209	0.896	0.957	0.984
LSIM [51]	2.344	1.156	0.835	1.175	0.815	0.943	0.975
PackNet-SfM [52]	2.343	1.158	0.887	1.234	0.821	0.945	0.968
GASDA [54]	1.356	1.156	0.963	1.223	0.765	0.897	0.968
VOMonodepth [53]	2.456	1.192	0.985	1.234	0.756	0.884	0.965
DeepV2D [50]	0.061	0.094	0.403	0.026	0.956	0.989	0.996

Note that, some of the methods in Table 5 such as monodepth2 [42] and PackNet-SfM [52] are only trained and evaluated on KITTI-ES(RD) as reported in their original papers. To achieve a fair and generalized comparison, we evaluated LSIM [51], PackNet-SfM [52], GASDA [54], VOMonodepth [53] and monodepth2 [42] on the NYU-v2 dataset [29]. The numbers for the rest of the methods are reported by the respective authors.

Table 6 compares the performances of the studied methods in terms of inference time. As shown in Table 6, BTS [49] has the fastest inference time with 0.22 s.

Table 6. Comparison of the models in terms of inference time (FC: fully convolutional; CNN: convolutional neural networks). Best method is emboldened and highlighted in green.

Method	Inference Time	Network/FC/CNN
BTS [49]	0.22 s	Encoder-decoder
VNL [48]	0.25 s	Auto-decoder
DeepV2D [50]	0.36 s	CNN
ACAN [46]	0.89 s	Encoder-decoder
VOMonodepth [53]	0.34 s	CNN
LSIM [51]	0.54 s	CNN
GASDA [54]	0.57 s	Encoder-decoder
DenseDepth [47]	0.35 s	Encoder-decoder
monoResMatch [38]	0.37 s	CNN
EMDEOM [32]	0.63 s	FC
DORN [18]	0.98 s	Encoder-decoder
PackNet-SfM [52]	0.97 s	CNN
monodepth2 [42]	0.56 s	CNN

An additional set of methods are studied and compared as presented in Appendix A. These methods are evaluated on either KITTI [31] or NYU-v2 [29] datasets and the comparison includes the parameter counts, depth accuracy measured using RMSE metric, memory requirement and training environment. All the methods in Appendix A, Table A1 are compared with the state-of-the-art monocular depth estimation methods. These methods are categorized as of low accuracy with expensive computational time and slow convergence rate which led us to exclude them from this survey.

Due to the technical complications with the publicly available codes and lack of instructions, we were not able to test all 13 methods for qualitative comparisons. Only five methods were implemented successfully and validated on NYU-v2 [29] dataset. A few samples of the results are illustrated in Figure 1. This visual comparison also supports the claim from the previous tables that DeepV2D [50] marginally outperforms BTS [49] and other methods as it can estimate smoother depth maps with sharper boundaries, less artifacts and relative scale.

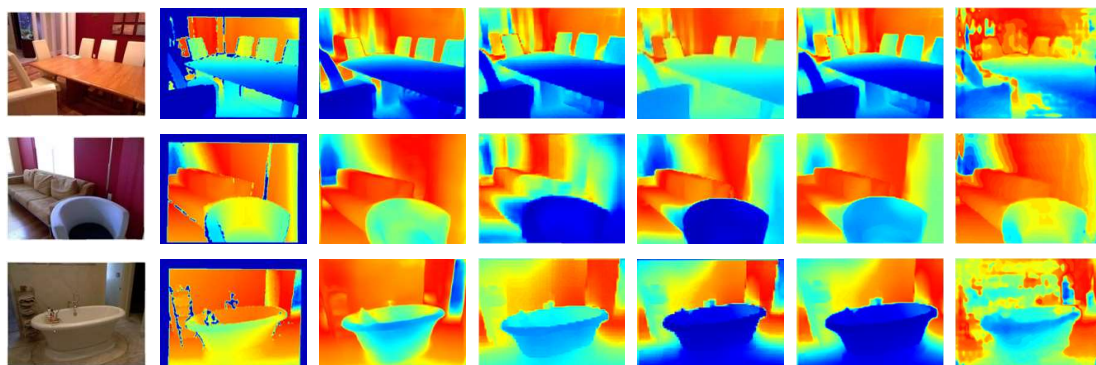


Figure 1. Qualitative comparison of five state-of-the-art-monocular depth estimation methods. From left to right: Input Image, Ground Truth, BTS [49], DeepV2D [50], DenseDepth [47], MonoResMatch [38] and DORN [18].

5. Discussion

Monocular depth estimation plays a crucial role in understanding 3D scene geometry in many applications. A single 2D image may be produced from an infinite number of distinct 3D scenes, which is a classical monocular depth estimation approach. The classical monocular depth estimation methods utilise meaningful monocular cues, such as perspective and texture information, objects size, object locations and occlusions, resulting in an undesirable low-resolution depth prediction. Recently, deep learning methods significantly improved the performance of the monocular depth estimation

methods by exploring image-level information and hierarchical features in the network. However, these methods employ repeated spatial pooling operations. To obtain high-resolution depth maps, skip connection-based networks are required, however, these methods tend to make the training process complicated and require more computational time. To target these issues, CNN based transfer learning methods were employed resulting in high-quality depth estimation. In general, deep-learning methods achieved outstanding results, however, they require a large amount of data labelled with precise depth measurements for training. The introduction of different methodologies and architectures such as local planar guidance layers (LPGL), multi-layer deconvolutional networks and atrous spatial pyramid have moved the performance of these models to the next level.

5.1. Comparison Analysis Based on Performance

I. Degree of supervision: most of the methods demonstrated in this paper require ground truth depth images for training. These supervised methods perform well and most of them are state-of-the-art on common benchmarks. Methods such as DeepV2D [50], BTS [49] and VNL [48] showed a much faster performance time compared to the other models. On the other hand, VNL [48], ACAN [46] and EMDEOM [32] provides the depth information with much lower resolution compared to the state-of-the-art. Unlike VNL [48], DORN [18] has the highest number of parameters in the supervised category and it requires a high number of operations making it an inefficient choice for real-life applications.

Obtaining large datasets of RGB images with accurate ground truth depth images is a challenging task. As such, methods that do not require full supervision (labelled ground truth) are more attractive. Methods such as LISM [51], monoResMatch [38], PackNet-SfM [52] and monodepth2 [42] are self-supervised methods. Although most of these methods can generate high resolution depth maps with comparable accuracy against the state-of-the-art, they are computationally expensive and require a significant amount of memory.

II. Accuracy and depth range: based on our evaluations, DeepV2D [50] marginally achieved the best performance compared to BTS [49] and the rest of the methods. On KITTI [31] dataset the model achieved 2.005 RMSE and threshold accuracy of 0.977 with $\delta < 1.25^3$. On NYUD-v2 [29] dataset it achieved 0.403 RMSE and threshold accuracy of 0.996 with $\delta < 1.25^3$. As shown in Tables 4 and 5, methods with 3D geometry constraint or features, outperform the others, which shows the importance of high order 3D geometric constraints for depth estimation.

The evaluation of BTS [49], DORN [18], VNL [48], DenseDepth [47] and VOMonodepth [53] indicated that supervised learning approaches achieved better results compared to semi and self-supervised methods.

III. Computation time and memory: based on the comparisons presented in Tables 3–6, VNL [48] significantly reduced the computational time and memory footprint, which can be used for both quality and low-cost monocular depth estimation.

The advancement of deep-learning methodologies suggests that cameras may become a competitive source of reliable 3D information. Compared to the conventional method, these models have the potential to be optimised for deployment on smart and consumer platforms.

These methods are composed in two ways: feature extraction which is done in encoder part using the powerful pre-trained models such as VGG [67], ResNet [57] or DenseNet [61], while the desired depth prediction is obtained using the decoder network architecture.

5.2. Future Research Directions

Over the past couple of years, deep-learning approaches have shown a significant improvement in the performance of monocular depth estimation. The topic is still in its infancy and further developments are yet to be expected. In this section, we present some of the current directions and issues for further future research.

1. Complex deep networks are very expansive in terms of memory requirements, which is a major issue when dealing with high-resolution images and when aiming to predict high-resolution depth images.
2. Developments in high-performance computing can address the memory and computational issues, however, devolving lighter deep network architectures remains desirable especially if it is to be deployed in smart consumer devices.
3. Another challenge is how to achieve higher accuracy, in general, which is affected by the complex scenarios, such as occlusions, highly cluttered scenes and complex material properties of the objects.
4. Deep-learning methods rely heavily on the training datasets annotated with ground truth labels for depth estimation which is very expensive to obtain in the real world.
5. We expect in the future to see the emergence of large databases for 3D reconstruction. Emerging new self-adoption methods that can adapt themselves to new circumstances in real-time or with minimum supervision are one of the promising future directions for research in depth estimation.

This paper provided a preliminary review of the recent developments in monocular depth estimation using deep-learning models. Regardless of its infancy, these methods are achieving promising results, and some of these methods are competing, in terms of accuracy of the results, with the traditional methods. We have entered a new era where deep learning and data-driven techniques play an important role in image-based depth estimation.

Author Contributions: Formal analysis, investigation, methodology and first draft by F.K.; Validation, review and editing the draft by S.S.; Supervision, validation, project administration and final draft preparation by H.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded under the fellowship award granted by the School of Engineering at National University of Ireland Galway.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Low-Performance Monocular Depth Estimation Methods

Table A1 summarizes the monocular depth estimation methods in terms of parameter counts, depth accuracy measured using RMSE metric, memory requirement and training environment. These methods are categorized as low accuracy with slow convergence rate and are excluded from this survey. All the numbers presented in this table are reported by the respective authors.

Table A1. Properties of the low-accuracy methods trained on either KITTI or NYU-v2 datasets. (FC: fully convolutional, ED: encoder-decoder, AD: auto-decoder, K: trained on KITTI dataset, N: trained on NYU-v2 dataset and CNN: convolutional neural networks).

Method	Input	Type	Optimizer	Parameters	Output	GPU Memory	RMSE	GPU Model
Zhou et al. [70]	128 × 416 K	CNN	Adam	N/A	128 × 416 K	N/A	4.975	N/A
Casser et al. [73]	128 × 416 K	CNN	Adam	N/A	128 × 416 K	11 GB	4.7503	1080 Ti
Guizilini et al. [74]	640 × 192 K	FC	Adam	86M	640 × 192 K	N/A	4.601	N/A
Godard et al. [15]	640 × 192 K	FC	Adam	31M	640 × 192 K	12 GB	4.935	TITAN Xp
Eigen et al. [33]	640 × 184 K	CNN	Adam	N/A	640 × 184	6 GB	N/A	TITAN Black
Guizilin et al. [75]	640 × 192 K	ED	Adam	79M	640 × 192	8 × 16 GB	4.270	Tesla V100
Tang et al. [76]	640 × 192 K	CNN	RMSprop	80M	640 × 192	12 GB	N/A	N/A
Ramamonjisoa et al. [40]	640 × 480 N	ED	Adam	69M	640 × 480 N	11 GB	0.401	1080 Ti
Riegler et al. [39]	N/A	ED	Adam	N/A	N/A	N/A	N/A	N/A
Ji et al. [37]	320 × 240 N	ED	Adam	N/A	320 × 240 N	12 GB	0.704	TITAN Xp
Almalioglu et al. [77]	128 × 416 K	GAN	RMSprop	63M	128 × 416 K	12 GB	5.448	TITAN V
Pillai et al. [41]	128 × 416 K	CNN	Adam	97M	128 × 416 K	8 × 16 GB	4.958	Tesla V100
Wofk et al. [24]	224 × 224 N	ED	SGD	N/A	224 × 224 N	N/A	0.604	N/A
Watson et al. [78]	128 × 416 K	ED	SGD	N/A	128 × 416 K	N/A	N/A	N/A
Chen et al. [79]	256 × 512 K	ED	Adam	N/A	256 × 512 K	11 GB	3.871	1080 Ti
Lee et al. [80]	640 × 480 N	CNN	SGD	61M	640 × 480 N	N/A	0.538	N/A

References

1. Chen, L.; Tang, W.; John, N.W.; Wan, T.R.; Zhang, J.J. Augmented Reality for Depth Cues in Monocular Minimally Invasive Surgery. *arXiv Prepr.* **2017**, arXiv:1703.01243.
2. Liu, X.; Sinha, A.; Ishii, M.; Hager, G.D.; Reiter, A.; Taylor, R.H.; Unberath, M. Dense Depth Estimation in Monocular Endoscopy with Self-supervised Learning Methods. *IEEE Trans. Med. Imaging* **2019**, *1*. [[CrossRef](#)] [[PubMed](#)]
3. Palafox, P.R.; Betz, J.; Nobis, F.; Riedl, K.; Lienkamp, M. SemanticDepth: Fusing Semantic Segmentation and Monocular Depth Estimation for Enabling Autonomous Driving in Roads without Lane Lines. *Sensors* **2019**, *19*, 3224. [[CrossRef](#)] [[PubMed](#)]
4. Laidlow, T.; Czarnowski, J.; Leutenegger, S. DeepFusion: Real-Time Dense 3D Reconstruction for Monocular SLAM using Single-View Depth and Gradient Predictions. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4068–4074.
5. Schönberger, J.L.; Frahm, J.M. Structure-from-Motion Revisited. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4104–4113.
6. Dai, Y.; Li, H.; He, M. Projective Multiview Structure and Motion from Element-Wise Factorization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2238–2251. [[CrossRef](#)]
7. Yu, F.; Gallup, D. 3D Reconstruction from Accidental Motion. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3986–3993.
8. Javidnia, H.; Corcoran, P. Accurate Depth Map Estimation From Small Motions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2453–2461.
9. Basha, T.; Avidan, S.; Hornung, A.; Matusik, W. Structure and motion from scene registration. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1426–1433.
10. Scharstein, D.; Pal, C. Learning conditional random fields for stereo. In Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), Minneapolis, MN, USA, 18–23 June 2007; pp. 1–8.
11. Scharstein, D.; Szeliski, R. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *Int. J. Comput. Vis.* **2002**, *47*, 7–42. [[CrossRef](#)]
12. Heikkila, J.; Silven, O. A four-step camera calibration procedure with implicit image correction. In Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), San Juan, PR, USA, 17–19 June 1997; pp. 1106–1112.
13. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [[CrossRef](#)]
14. Javidnia, H.; Corcoran, P. A Depth Map Post-Processing Approach Based on Adaptive Random Walk with Restart. *IEEE Access* **2016**, *4*, 5509–5519. [[CrossRef](#)]
15. Godard, C.; Aodha, O.M.; Brostow, G.J. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6602–6611.
16. Kuznietsov, Y.; Stückler, J.; Leibe, B. Semi-Supervised Deep Learning for Monocular Depth Map Prediction. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2215–2223.
17. Bazrafkan, S.; Javidnia, H.; Lemley, J.; Corcoran, P. Semiparallel deep neural network hybrid architecture: First application on depth from monocular camera. *J. Electron. Imaging* **2018**, *27*, 1–19. [[CrossRef](#)]
18. Fu, H.; Gong, M.; Wang, C.; Batmanghelich, K.; Tao, D. Deep ordinal regression network for monocular depth estimation. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2002–2011.
19. Xu, D.; Wang, W.; Tang, H.; Liu, H.; Sebe, N.; Ricci, E. Structured attention guided convolutional neural fields for monocular depth estimation. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3917–3925.
20. Microsoft. Kinect for Windows. 2010. Available online: <https://developer.microsoft.com/en-us/windows/kinect/> (accessed on 22 March 2020).

21. Microsoft. Kinect for Xbox One. 2017. Available online: <https://www.xbox.com/en-US/xbox-one/accessories/kinect> (accessed on 22 March 2020).
22. Javidnia, H. *Contributions to the Measurement of Depth in Consumer Imaging*; National University of Ireland Galway: Galway, Ireland, 2018.
23. Elkerdawy, S.; Zhang, H.; Ray, N. Lightweight monocular depth estimation model by joint end-to-end filter pruning. In Proceedings of the 2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, 22–25 September 2019; pp. 4290–4294.
24. Wofk, D.; Ma, F.; Yang, T.-J.; Karaman, S.; Sze, V. Fastdepth: Fast monocular depth estimation on embedded systems. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 6101–6108.
25. Poggi, M.; Aleotti, F.; Tosi, F.; Mattoccia, S. Towards real-time unsupervised monocular depth estimation on cpu. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, 1–5 October 2018; pp. 5848–5854.
26. Scharstein, D.; Szeliski, R. High-accuracy stereo depth maps using structured light. In Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), Madison, WI, USA, 16–22 June 2003; Volume 1, pp. I-195–I-202.
27. Luo, H.; Gao, B.; Xu, J.; Chen, K. An approach for structured light system calibration. In Proceedings of the 2013 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, Nanjing, China, 26–29 May 2013; pp. 428–433.
28. Velodyne Lidar. Inc. Available online: <https://velodynelidar.com/> (accessed on 22 March 2020).
29. Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor segmentation and support inference from rgb-d images. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 746–760.
30. Saxena, A.; Sun, M.; Ng, A.Y. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *31*, 824–840. [[CrossRef](#)]
31. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
32. Dos Santos Rosa, N.; Guizilini, V.; Grassi, V. Sparse-to-Continuous: Enhancing Monocular Depth Estimation using Occupancy Maps. In Proceedings of the 2019 19th International Conference on Advanced Robotics (ICAR), Belo Horizonte, Brazil, 2–6 December 2019; pp. 793–800.
33. Eigen, D.; Puhersch, C.; Fergus, R. Depth Map Prediction from a Single Image Using a Multi-scale Deep Network. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2366–2374.
34. Uhrig, J.; Schneider, N.; Schneider, L.; Franke, U.; Brox, T.; Geiger, A. Sparsity Invariant CNNs. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 11–20.
35. Borghi, G.; Venturelli, M.; Vezzani, R.; Cucchiara, R. Poseidon: Face-from-depth for driver pose estimation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4661–4670.
36. Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; Brox, T. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4040–4048.
37. Ji, R.; Li, K.; Wang, Y.; Sun, X.; Guo, F.; Guo, X.; Wu, Y.; Huang, F.; Luo, J. Semi-Supervised Adversarial Monocular Depth Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *1*. [[CrossRef](#)]
38. Tosi, F.; Aleotti, F.; Poggi, M.; Mattoccia, S. Learning monocular depth estimation infusing traditional stereo knowledge. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9799–9809.
39. Riegler, G.; Liao, Y.; Donne, S.; Koltun, V.; Geiger, A. Connecting the Dots: Learning Representations for Active Monocular Depth Estimation. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7624–7633.

40. Ramamonjisoa, M.; Lepetit, V. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. In Proceedings of the 2019 IEEE International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019.
41. Pillai, S.; Ambruş, R.; Gaidon, A. Superdepth: Self-supervised, super-resolved monocular depth estimation. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 9250–9256.
42. Godard, C.; Mac Aodha, O.; Firman, M.; Brostow, G.J. Digging into self-supervised monocular depth estimation. In Proceedings of the 2019 IEEE International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 3828–3838.
43. Chen, Y.; Zhao, H.; Hu, Z. Attention-based context aggregation network for monocular depth estimation. *arXiv Prepr.* **2019**, arXiv:1901.10137.
44. Alhashim, I.; Wonka, P. High quality monocular depth estimation via transfer learning. *arXiv Prepr.* **2018**, arXiv:1812.11941.
45. Yin, W.; Liu, Y.; Shen, C.; Yan, Y. Enforcing geometric constraints of virtual normal for depth prediction. In Proceedings of the 2019 IEEE International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 5684–5693.
46. Lee, J.H.; Han, M.-K.; Ko, D.W.; Suh, I.H. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv Prepr.* **2019**, arXiv:1907.10326.
47. Teed, Z.; Deng, J. Deepv2d: Video to depth with differentiable structure from motion. *arXiv Prepr.* **2018**, arXiv:1812.04605.
48. Goldman, M.; Hassner, T.; Avidan, S. Learn stereo, infer mono: Siamese networks for self-supervised, monocular, depth estimation. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.
49. Guizilini, V.; Ambrus, R.; Pillai, S.; Gaidon, A. Packnet-sfm: 3d packing for self-supervised monocular depth estimation. *arXiv Prepr.* **2019**, arXiv:1905.02693.
50. Andraghetti, L.; Myriokefalitakis, P.; Dovesi, P.L.; Luque, B.; Poggi, M.; Pieropan, A.; Mattocchia, S. Enhancing self-supervised monocular depth estimation with traditional visual odometry. In Proceedings of the 2019 International Conference on 3D Vision (3DV), Quebec City, QC, Canada, 16–19 September 2019; pp. 424–433.
51. Zhao, S.; Fu, H.; Gong, M.; Tao, D. Geometry-aware symmetric domain adaptation for monocular depth estimation. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9788–9798.
52. Chen, W.; Fu, Z.; Yang, D.; Deng, J. Single-image depth perception in the wild. In Proceedings of the Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, Barcelona, Spain, 5–10 December 2016; pp. 730–738.
53. Xie, J.; Girshick, R.; Farhadi, A. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 842–857.
54. Garg, R.; BG, V.K.; Carneiro, G.; Reid, I. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 740–756.
55. Ramos, F.; Ott, L. Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent. *Int. J. Rob. Res.* **2016**, *35*, 1717–1730. [[CrossRef](#)]
56. Laina, I.; Rupprecht, C.; Belagiannis, V.; Tombari, F.; Navab, N. Deeper Depth Prediction with Fully Convolutional Residual Networks. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 239–248.
57. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
58. Liu, W.; Rabinovich, A.; Berg, A.C. Parsenet: Looking wider to see better. *arXiv Prepr.* **2015**, arXiv:1506.04579.
59. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7794–7803.

60. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
61. Huang, G.; Liu, Z.; Van Der, M.L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
62. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Li, F.-F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, FL, USA, 20–25 June 2009.
63. Lehtinen, J.; Munkberg, J.; Hasselgren, J.; Laine, S.; Karras, T.; Aittala, M.; Aila, T. Noise2noise: Learning image restoration without clean data. *arXiv Prepr.* **2018**, arXiv:1803.04189.
64. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5828–5839.
65. Chen, L.-C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv Prepr.* **2017**, arXiv:1706.05587.
66. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 850–865.
67. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv Prepr.* **2014**, arXiv:1409.1556.
68. Huber, P.J. Robust Estimation of a Location Parameter. In *Breakthroughs in Statistics: Methodology and Distribution*; Kotz, S., Johnson, N.L., Eds.; Springer New York: New York, NY, USA, 1992; pp. 492–518.
69. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
70. Zhou, T.; Brown, M.; Snavely, N.; Lowe, D.G. Unsupervised learning of depth and ego-motion from video. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1851–1858.
71. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III*; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
72. Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
73. Casser, V.; Pirk, S.; Mahjourian, R.; Angelova, A. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. Proceedings of Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19), Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8001–8008.
74. Guizilini, V.; Hou, R.; Li, J.; Ambrus, R.; Gaidon, A. Semantically-Guided Representation Learning for Self-Supervised Monocular Depth. *arXiv Prepr.* **2020**, arXiv:2002.12319.
75. Guizilini, V.; Li, J.; Ambrus, R.; Pillai, S.; Gaidon, A. Robust Semi-Supervised Monocular Depth Estimation with Reprojected Distances. *arXiv Prepr.* arXiv:1910.01765, 2019.
76. Tang, C.; Tan, P. Ba-net: Dense bundle adjustment network. *arXiv Prepr.* **2018**, arXiv:1806.04807.
77. Almalioglu, Y.; Saputra, M.R.U.; Gusmão PPBd Markham, A.; Trigoni, N. GANVO: Unsupervised Deep Monocular Visual Odometry and Depth Estimation with Generative Adversarial Networks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5474–5480.
78. Watson, J.; Firman, M.; Brostow, G.J.; Turmukhambetov, D. Self-Supervised Monocular Depth Hints. In Proceedings of the 2019 IEEE International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 2162–2171.

79. Chen, P.-Y.; Liu, A.H.; Liu, Y.-C.; Wang, Y.-C.F. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 2624–2632.
80. Lee, J.-H.; Kim, C.-S. Monocular depth estimation using relative depth maps. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9729–9738.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Appendix B

A Review of Benchmark Datasets and Training Loss Functions in Neural Depth Estimation

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

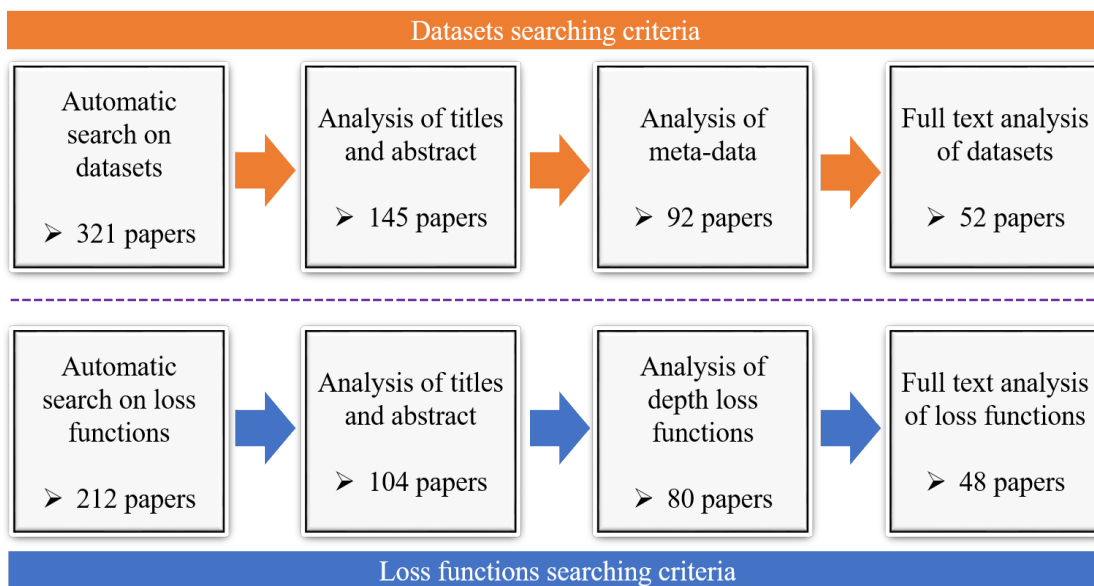


FIGURE 1. An illustration of the methodology adopted for conducting the survey categorizes depth estimation databases and the loss functions.

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

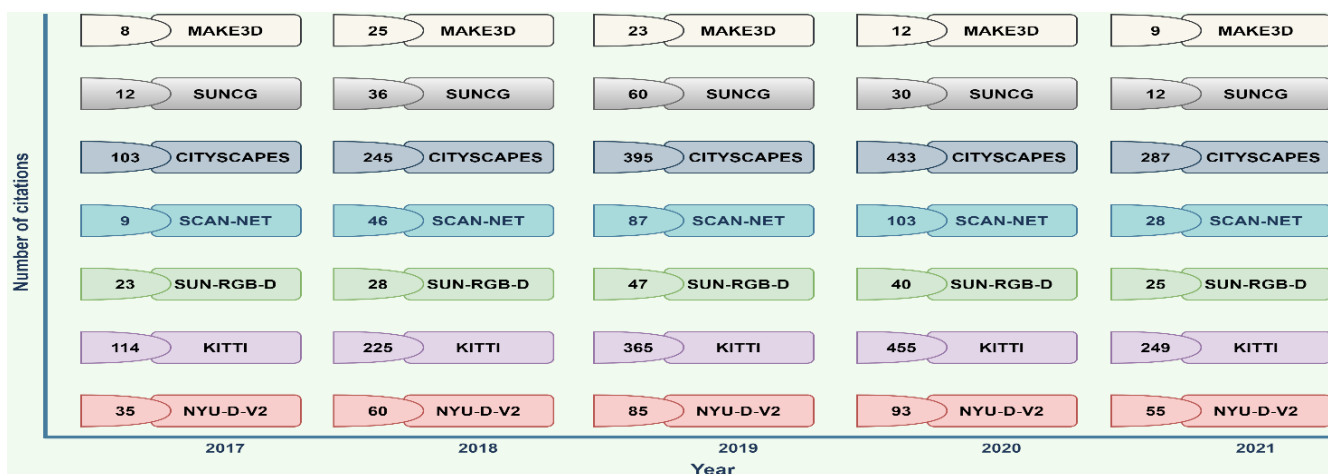


FIGURE 2. Mag an illustration of database according to the number of citations in each year from 2017 to 2021. The number against each database represents the total citations in each year.

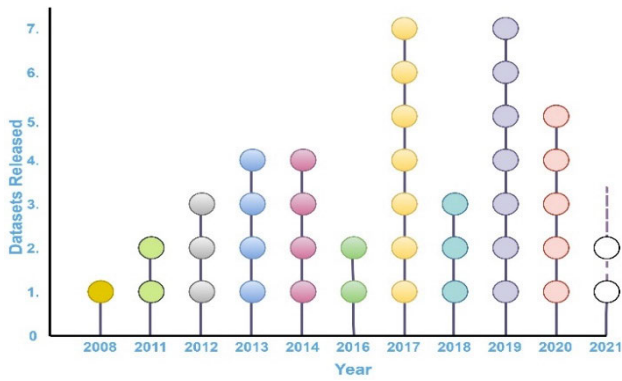


FIGURE 3. The amount of depth datasets released each year, with predicted releases in 2021 represented as a dashed line.

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.

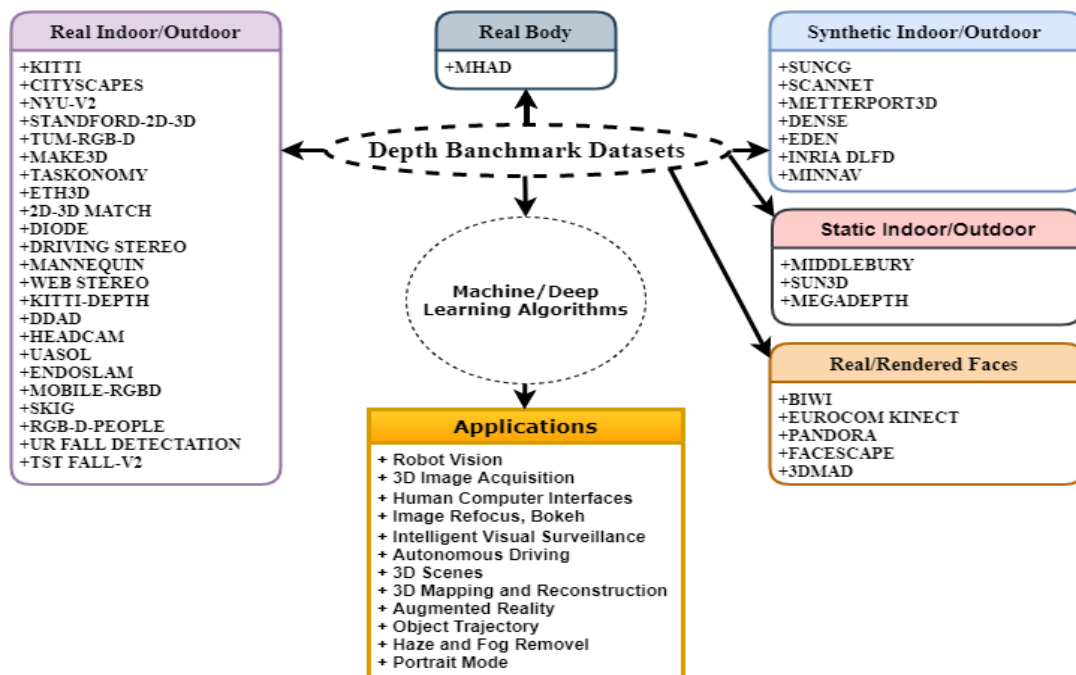


FIGURE 4. Organized classification of depth datasets studied in this paper, which shows different use case applications of each categories.

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 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:

$$L_2(y, \check{y}) = \frac{1}{N} \sum_i^N (y_i - \check{y}_i)^2 \quad (1)$$

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. Snaveley, 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-Carri6, 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. UmmeHofer, 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. UmmeHofer, 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*.

...

Appendix C

Towards Monocular Neural Facial Depth Estimation: Past, Present, and Future

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Towards Monocular Neural Facial Depth Estimation: Past, Present, and Future

FAISAL KHAN¹, MUHAMMAD ALI FAROOQ¹, WASEEM SHARIFF¹, SHUBHAJIT BASAK², AND PETER CORCORAN¹ (Fellow, IEEE)

¹Department of Electronic Engineering, College of Science and Engineering, National University of Ireland Galway, Galway, H91 TK33 Ireland

²School of Computer Science, National University of Ireland Galway, Galway H91 TK33, Ireland

Corresponding author: Faisal Khan (f.khan4@nuigalway.ie).

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.”

ABSTRACT This article contains all of the information needed to conduct a study on monocular facial depth estimation problems. A brief literature review and applications on facial depth map research were offered first, followed by a comprehensive evaluation of publicly available facial depth datasets and widely used loss functions. The key properties and characteristics of each facial depth map dataset are described and evaluated. Furthermore, facial depth maps loss functions are briefly discussed, which will make it easier to train neural facial depth models on a variety of datasets for both short- and long-range depth maps. The network's design and components are essential, but its effectiveness is largely determined by how it is trained, which necessitates a large dataset and a suitable loss function. Implementation details of how neural depth networks work and their corresponding evaluation matrices are presented and explained. In addition, an SoA neural model for facial depth estimation is proposed, along with a detailed comparison evaluation and, where feasible, direct comparison of facial depth estimation methods to serve as a foundation for a proposed model that is utilized. The model employed shows better performance compared with current state-of-the-art methods when tested across four datasets. The new loss function used in the proposed method helps the network to learn the facial regions resulting in an accurate depth prediction. The network is trained on synthetic human facial depth datasets whereas for validation purposes real as well as synthetic facial images are used. The results prove that the trained network outperforms current state-of-the-art networks performances, thus setting up a new baseline method for facial depth estimations.

INDEX TERMS Facial depth datasets, Loss functions, Neural depth estimation, Empirical and systematic evaluation

I. INTRODUCTION

The process of obtaining 3D information from a 2D frame is known as depth estimation. Depth estimation is used in diversified computer vision applications such as augmented reality, posture estimation, 3D reconstruction, object detection and recognition, semantic segmentation and -human-machine interaction, weather forecast, and autonomous vehicles. The ground truth depth information used to estimate depth is beneficial for developing reliable navigation systems for intelligent vehicles, environmental reconstruction, and image interpretation to understand the objects in the image and the scene behind them.

Face depth estimation is a challenging subject that has been explored in conjunction with face motion [1], facial analysis, and facial recognition [2], [3]. Many methods for estimating face depth have been presented in recent years, notably 3D from stereo replicating [4], 3D morphable model-based methods [5], [6], shape from shading (SfS) [5, 6], shape from motion techniques (SfM) [6], [7], and statistical techniques [8], [9]. Due to the facial symmetry of facial areas, the stereo matching procedure for face depth estimation is more complicated (regardless of utilizing the local or global technique), particularly when the system is binocular and therefore only one stereo pair is used. Stereo matching methods can estimate a reasonable depth or disparity map for

facial depth estimation, but these approaches are more sophisticated, requiring the use of a local or global procedure. Because of the similarity of the face areas, particularly when using a binocular setup with only one pair of stereo images. All stereo approaches are limited by the similarity characteristics of the facial information. Furthermore, the similarity of the pixels values results in more spikes, holes, and particularly uncertain disparities in the depth map.

The computer vision field has conventionally approached the field of depth maps in a variety of methods, such as with stereo or multi-view cameras [10], [11], structure from motion [12], [13], and depth from light diffusion & shading [14], [15]. The described methods face many difficulties, such as missing pixel values and depth consistency, which result in inconsistencies in depth maps. In addition, the camera calibration, camera setup, and post-processing techniques are computationally expensive and time-consuming. The research community has explored the monocular depth estimation task using only a single image which is much more straightforward and suitable for consumer applications. The credit goes to significant advances in machine learning-based networks [16]–[20]. In the first part of the paper, we have given a detailed evaluation of publicly available facial depth datasets and widely used loss functions in facial depth estimation networks, thus to better understand the problem of facial depth maps. The key characteristics and properties of the facial depth datasets are presented and compared, followed by the loss functions employed. The implementation specifics of how neural depth networks work, as well as the evaluation matrices that correlate to them, are shown and described. A full comparison evaluation and, where possible, direct comparison of facial depth estimation methods are performed in the second phase of the paper to serve as a foundation for a proposed model that is used. When tested across four datasets, the proposed model outperforms current state-of-the-art approaches. The suggested method's unique loss function aids the network in learning the facial areas, resulting in an accurate depth prediction. The network is trained using synthetic human facial depth datasets, and real and synthetic facial images from four facial depth datasets are used for validation.

A. RESEARCH CONTRIBUTIONS

Following thorough research over the previous few years, image-based facial depth estimation using deep learning algorithms has demonstrated promising results. However, the field is still in its early stages, and more improvements are expected to address issues and challenges such as data selection for training, generalization to unknown environments, fine-scale depth estimation, reconstruction versus recognition, handling multiple objects in the presence of occlusions, and cluttered backgrounds, data imbalance and how to select an appropriate loss function and neural model for facial depth estimation.

This paper aims to provide all of the key information for conducting a study on monocular facial depth estimation challenges. First, a brief review of the literature and applications of facial depth map research was presented, followed by a detailed analysis of publicly available facial depth datasets and commonly used loss functions. To better understand the facial depth map problem, the facial depth dataset's key characteristics and properties are described and evaluated, followed by the loss functions used. For each dataset, the dataset description, metadata, ground truth, and relevant data (year of publishing, ground truth information, image size, type, objects per image, and several images) are listed systematically. In addition, each loss function is presented in such a way that the research community can select the best loss function for their requirements. The implementation details of how neural depth networks work are demonstrated and explained, as are the evaluation matrices that correspond to them. In the second section of the paper, a complete comparison evaluation and, where possible, direct comparison of facial depth estimation methods are conducted to serve as a foundation for a proposed model that is used. The model outperforms current state-of-the-art techniques when tested across four datasets. The unique loss function of the suggested method supports the network in learning the facial areas, resulting in an accurate depth prediction. The network is trained with synthetic human facial depth datasets and validated with real and synthetic facial images from four facial depth datasets.

The following is how the rest of the paper is organized: Section 2 discusses related work in the domain of facial depth estimation, especially related studies, or surveys. Section 3 presents the results of a bibliometric investigation, a thorough examination of depth datasets, and further discusses the most used loss functions. Section 4 presents the implementation details of how facial depth neural networks work followed by some comparative analysis of the facial depth estimation methods. Section 5 presents evaluation matrices and section 6 describes and illustrates the most recent SoA depth estimation model, which is discussed and chosen for facial depth estimation. Section 7 shows the experimental results, discusses the training approach, and compares the trained model to SoA methods in a brief comparison study. Section 8 includes a detailed discussion of the experimental results while section 9 provides the conclusion and future research directions.

II. RELATED WORKS

Datasets are the foundations for evaluating the behaviour and validating the results of artificial intelligence networks, and they play a critical role in scientific research. Another important building block is to use an appropriate loss function to improve the deep network's training performance. An in-depth analysis of various facial depth datasets is performed, and depth regression loss functions for both short and long-range depth datasets are proposed in the next sections. This

section focuses mostly on related facial depth estimation research and applications.

A. FACIAL DEPTH ESTIMATION APPLICATIONS

Human face images are among the most common images, and they play an important role in many visual interpretations. Since the facial parts separation in a human face is well-known in human anthropometry, it is possible to find the distance of a human focus from a single image frame with good accuracy provided an understanding of the camera's field-of-view. The research community in today's fast-paced technological environment wants more realistic representations, thus 3D representations of 2D images are becoming increasingly important. These methods are categorized into the following primary categories based on their applications.

Feature Extraction Methods: The expressions on people's faces reveal information about individuals. Faces identify people, and one may infer how others are feeling from their expressions. Face feature extraction can help in the improvement of face depth maps tasks. In the realm of computer vision, facial feature depth estimation and 3D reconstruction are popular topics. In computer vision related applications such as detection and recognition, especially under shifting posture lighting, and expression, 3D information gives significant benefits in overcoming difficulties associated with 2D images (PIE) [14]. Methods have been shown in the SoA to be a potential solution to several of problems in facial depth maps [20], [21], [22], [23], [24], [25].

Feature Fusion Methods: Feature fusion offers a full description of image features' rich internal information, and following dimensionality reduction, compact representations of integrated features can be obtained, resulting in decreased computational complexity and better performance of facial depth maps. 3D reconstruction helps in the resolution of difficulties in 2D images as well as the improvement in performance in a variety of tasks. Several approaches have been offered in the last few years [26], [27], [28], [29], [30], [31], [32], [33], [34] for facial depth estimation tasks.

Image Processing Filtration Methods: For the successful application of depth information, quality is critical. Visually undesirable rendered views are frequently produced when a depth map is distorted by large featureless artifacts. A robust depth image post-filtering technique should be considered for further 3D video transmission. Filtering of depth maps has primarily been studied from the viewpoint of increasing resolution [35], [36], [37]. There are a variety of post-processing techniques for restoring natural images [38]. Filtering algorithms included Gaussian smoothing and the H.264 in-loop deblocking filter [39], as well as a local polynomial approximation (LPA) [40] and bilateral filtering [41], which use edge-preserving structure information from the colour channel to refine rough depth maps [42].

Table I shows the corresponding methods categorized into feature extraction, feature fusion, and image processing filtration with their respective use cases and strategies involved.

TABLE I: Properties of feature, fusion, and image processing filtration methods

Method Category	Methods	Strategy	Category	Descriptions of the main block	Uses
Feature Extraction	[14] [20] [22] [23] [25]	Depth From Shading, Defocus Face Depth CNN Recovering Facial Shape Shape-From-Shading From Depth Maps CNN	DL DL ML ML DL	Light-Field Angular Function Adversarial Networks Surface Normal Direction Symmetric Self-Ratio Images Feature Extractor	Depth Maps Depth Maps Reconstructions Reconstructions Object Recognition
Feature Fusion	[26] [27] [28] [29] [30] [31] [32] [33] [34]	Face Depth CNN Face Depth CNN Autoencoder Single Facial Depth Map Face From Depth Face From Depth Pose 3D Blendshape Learning Feature	DL DL DL DL DL DL DL DL ML	Single Reference Face Shape Multi-Level Feature Fusion Stacked Contractive Autoencoder Multi-Level Feature Fusion Feature Fusion Extractor Feature Fusion Extractor Multi-Level Feature Fusion Feature Fusion Extractor Multi-Level Feature Fusion	3D Face Reconstruction 3D Reconstruction Learning 3D Faces Refinement Driver Pose Estimation Image Super-Resolution Pose Estimation Facial Expression Recognition Aggregation
Image Processing Filtration	[36] [37] [38] [39] [40] [41] [42]	Learning Feature Depth Pointwise Shape-Adaptive Pointwise Shape-Adaptive Local Approximation For Gray And Color Images Fused Deep Representation	ML ML ML ML ML DL DL	Joint Bilateral Multistep Joint Bilateral High-Quality Filtration Filters High-Quality Filtration Bilateral Filtering Light Field	Upsampling Depth Upsampling Denoising And Deblocking Deblocking Signal And Image Processing Signal And Image Processing Face Recognition

1) Facial depth in 3D face recognition

Face recognition (FR) has been used for human identification for ages. With the advances of deep neural

networks (DNNs), both face identification (one-to-many) and face verification (one-to-one) have achieved state-of-the-art results. Despite these advances, there are still a few

limitations due to external conditions like viewing angles, human appearances like facial expressions, occlusions, scene lightings. To overcome these factors researchers, use other modalities like depth and surface normal. The availability of low-cost RGB-D consumer level sensors like Microsoft Kinect and Intel Real Sense which simultaneously capture depth data of the scene and the colour intensity make these

multimodal data more accessible. Depth information can be very useful in FR because it helps to retrieve geometric information of the face in the form of dense 3D points. RGB-D FR can be categorized broadly into two classes – handcrafted feature-based method and deep learning-based methods. Table II shows the corresponding details of the listed methods for this subsection.

TABLE II: Properties of facial recognition depth maps methods

Methods	Feature Type	Features extracted	Strategy	Method Category	Descriptions of the main block	Uses
[43]	Geometric	Histogram Of Oriented Gradient (HOG)	Random Decision Forest (RDF) Classifier	Feature Extraction DL	Entropy Map	Recognition
[44]	Geometric	Local Binary Patterns (LBP)	Iterative Closest Point (ICP) And	Feature Extraction DL	Discriminant Color Space (DCS)	Depth Maps
[45]	Geometric	Signed Distance Function (SDF)	ICP	Feature Extraction ML	3D Face Model	Depth Maps
[46]	Statistical	Feature Space	CNN	Feature Fusion DL	Autoencoder	Depth Maps
[47]	Spatial	Feature Space	Single Facial Depth	Feature Fusion	CNN VGG	Depth Maps & Recognition
[48]	Spatial and Geometric	Feature Space	Face Recognition Accuracy	Feature Extraction	Surface Normal, Point Cloud;	Recognition & Depth Maps

B. FACIAL DEPTH FROM STEREO & MULTI-VIEW

Using two or more cameras, depth can be derived from stereo or multi-view. A process known as stereo matching is used to produce this map. The primary notion is that triangulation and stereo matching can be used to estimate depth in a variety of applications, including object grasping, collision avoidance, broadcasting, robotic navigation, and multimedia. The most frequently used methods for measuring face depth from stereo methods are designed on fitting the computed depth to a generalized 3D model [49], [50], [51]. For facial depth estimation, a passive stereo system for 3D human face reconstruction and recognition at a distance method is introduced [52]. Using a Kinect camera and a face detection algorithm, a method was able to reliably locate the human head and estimate head posture. To locate the detailed facial characteristics, a depth AAM algorithm is designed [53]. In a passive stereo vision system, a method for estimating facial depth is introduced. The method relies on the fast creation of facial disparity maps, which does not necessitate the use of expensive instruments or generic face models. It entails including face attributes in the disparity

estimate process to improve 3D face reconstruction [54].

The primary drawbacks of these approaches are the long processing times associated with the fitting phase (due to the high computational complexity) and the need for human setup, as seen in [51]. Another drawback of these approaches is that the generated faces resemble the generic model rather than their model. It's also particularly sensitive to noise because it calculates curves using the second derivative.

C. FACIAL DEPTH FROM 2D, MONOCULAR IMAGES

The monocular depth estimation method uses only a single RGB image as input to predict the depth value of each pixel or infer depth information. The following methods use a monocular depth strategy. Monocular depth maps are simple to set up, especially when it comes to camera calibration, and only require a single image to estimate depth. It can also give a variety of monocular visual cues, such as gradients and texture variations, colour, and defocus, that have previously been underutilized in such systems and can be used even in texture fewer areas. Table III shows the corresponding details of the listed methods from this section.

TABLE III: Properties of facial depth from 2D monocular images methods

Methods	Feature Type	Features extracted	Strategy	Method Category	Descriptions of the main block	Uses
[26]	Geometric	Single Reference Face Shape	Constrained Independent Component Analysis	Feature Extraction DL	3D Face Model	3D Face Reconstruction
[9]	Spatial & Geometric	Constrained Independent Component Analysis	The Rotation and Translation Process	Feature Extraction DL	Discriminant Color Space (DCS)	3D Face Reconstruction

[7]	Geometric	Similarity Transform & Feature Space	Deep Learning	Feature Extraction	3D Face Model	Depth Maps & 3D Face Reconstruction
[55]	Statistical	End-To-End Learning	Uses Single-View Depth and Multi-View Pose Networks	Feature Fusion	CNN Models Combined	Depth Maps
[56]	Spatial & Geometric	Canonical Correlation Analysis Surface Depth.	Surface Depth	Feature Extraction	Face Color Texture And Surface Depth	Face Depth Maps
[57]	Spatial & Geometric	Feature Points, Feature Space	Feature Points Similarity Analysis	Feature Extraction DL	Extracted To Form The 2D-3D	3D Face Reconstruction
[58]	Geometric	Recovering The Depth	Uses A Cascaded FCN And CNN Architecture	Feature Extraction	CNN Models Combined	Face Depth Estimation
[59]	Spatial & Geometric	Feature Space	Uses A Combination of Loss Function	Feature Extraction	CNN Encoder-Decoder	Face Depth Estimation

D. FACIAL DEPTH THROUGH DOMAIN TRANSLATION

The domain translation which is also known as image translation requires learning a parametric mapping function between two separate domains. Per-pixel classification or regression issues are frequently used to solve image-to-image translation challenges [48, 49, 50] [60], [61], [62]. [51] [30] suggested a method for computing the appearance of a face based on a standard CNN that combines characteristics of autoencoders and fully connected convolutional networks (FCN). Several recent studies have investigated the image-to-image translation problem by developing a mapping between two frames using conditional generative adversarial networks [52] [63]. Authors in [53] [64], proposed an approach with the pix2pix model, which synthesizes images from semantic labelling and then reconstructs objects from edges and colourizes images. [54] [65] provided a framework of linked GANs that can synthesize pairs of similar images in two separate contexts. This research also focuses on the domain translation problem to create visually attractive facial depth maps with sufficient discriminative information for face recognition.

The authors [66] present a novel framework for learning (1) RGB face parsing, (2) depth face parsing, and (3) RGB-to-depth domain translation together for facial depth maps. In [67], the authors suggest a new Deterministic Conditional GAN that is efficient for face-to-face translation from depth to RGB and is trained on labelled RGB-D face datasets. Whereas the network cannot reconstruct the exact somatic attributes of unknown focus on the individual, it can reconstruct plausible faces which is sufficient for use in various pattern recognition applications. In [68] a method proposes face from depth for head pose estimation on depth images for estimating head and shoulder pose based solely on depth images to create a complete end-to-end system. The proposed method also incorporates head detection and a localization module for facial depth estimation.

E. FACIAL DEPTH MAP DENOISING

Two forms of noise which include holes and spikes impact the depth data generated by the face reconstruction process. Pixels with unknown depth values are referred to as holes. During the disparity estimation procedure, the disparity values for these pixels are set to zero. They arise when there

is an obstruction or poor light. Spikes are pixels having an incorrect depth estimation. They are mostly caused by incorrect matching and occur inhomogeneous areas where pixels have similar intensity values.

Various approaches for face depth map de-noising have been presented in the literature. These methods are divided into two categories: global and local. To eliminate spikes and fill holes, global approaches apply noise reduction filters to the hole depth image. For this, the median filter is frequently used. Authors in [69], [70], proposed a Gaussian filter method that works to soften the data and eliminate spikes in the z-coordinate. To eliminate spikes, fill tiny gaps, and smooth the data, the authors in [71] utilized three median filters with different variances. For minor noises, these types of filters can produce optimal results. However, if the noisy region is big, these filters will not be able to remove the noise; instead, they will just modify the pixel values by their surrounding pixels.

In [49] by processing the data row by row, with the first and last non-zero pixels in each row being chosen by a sweep of the depth images. This procedure is continued until no more pixels are produced. The filling process usually involves utilizing an interpolation technique or a local median filter after determining the hole's boundaries. This method is more accurate than the global method since it just processes noises and leaves the non-noisy data alone. Since holes have a known value (zero or undefined), it can only handle those; spikes, on the other hand, have a random value, therefore it can't be used to eliminate them.

The authors [72] suggested an edge-guided deep neural network for the super-resolution of a single facial depth map. It is divided into two sub-networks: edge prediction and depth reconstruction. The edge prediction sub-network generates an edge guidance map that is used to guide the depth reconstruction sub-network in recovering sharp edges and fine constructions. [73] proposes a time-of-flight depth camera-specific wavelet-based depth video denoising approach based on multi hypothesis motion estimation for facial depth maps. In [74] authors proposed a method and system for super-solving and recovering the facial depth maps. The main idea of this approach is to use a learning-

based technique to gather reliable face priors from a high-quality facial depth map to improve the depth images.

III. PUBLICLY AVAILABLE FACIAL DEPTH ESTIMATION DATASETS AND LOSS FUNCTIONS

This section provides an overview of the most commonly used facial image depth datasets, including their respective descriptions in tabular form.

There are several useful datasets available for training depth estimation methods both multi-view and monocular images of human faces. The collection's general data contains information on the number of objects, scenarios, and RGB and depth images. Among the numerous types of data contained within every dataset, the ground truth contains depth, mesh, cameras trajectories, videos, positions, point cloud, semantics label, trajectories, and dense multi-class labelling. As the field of face image depth estimation research grows in popularity, more work is being put into creating higher and additional informative depth maps datasets. Fig. 1 shows the number of new publicly available facial depth maps datasets and their corresponding number of citations becoming available each year over the period for the last ten (10) years. Table 4-6 tabulates a comparison analysis for the data existing in each dataset.

A. FACIAL AND POSE DEPTH DATASETS

The depth camera sensor should be capable of faster human-skeletal tracking in addition to being a low-cost camera sensor that outputs both RGB and depth information. This kind of tracking can provide the precise position of human body joints

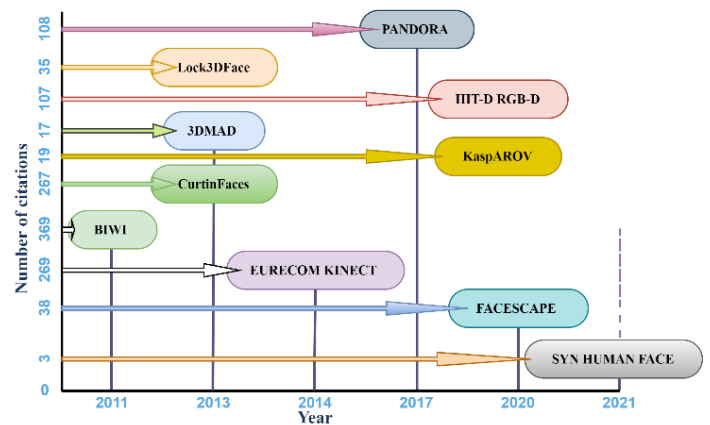


FIGURE 1. The number of depth datasets publicly available every year, with predicted availability in the year 2021 embodied as a dashed line.








throughout a period, making comprehensive human behaviour investigations easier and quicker. As a consequence, there has been a lot of interest in inferring human faces from depth images and synthesizing depth and RGB images. Several new facial depths maps datasets have been generated in recent years to assist in the confirmation of humanoid facemask action analysis methods. The details of these datasets are provided in the following section.



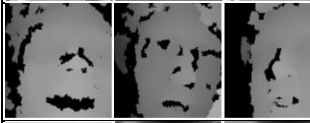
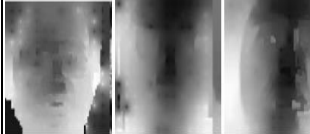
TABLE IV: Comparison between data representations

❖	RGB: Images of the visible light spectrum in two dimensions.
❖	Depth: The term "depth map" refers to a map of per-pixel data that includes depth-related information. The distance to an object at each pixel is specified by a depth map (e.g., distance from the camera).
❖	Video: This type of data displays a series of temporally consecutive visual readings.
❖	Point cloud: A 3-dimensional shape is represented by a collection of points, each of which has at least one x, y, and z coordinate.
❖	Mesh: It's a polygon-based representation of 3-dimensional objects that captures topological and shape surfaces directly.
❖	Scene: It's a form of data that are collected in a specific environment, such as a room or various indoor/outdoor scenarios.
❖	Semantic: Labels that relate some data to an ontology class (e.g., human, vehicle, etc.).
❖	Object: Object properties such as form, and motion are captured in data. appropriate for tasks such as tracking or object categorization.
❖	Camera: This information can be used to track the geometrical properties of the camera.
❖	Action: This information is made up of videotapes of people performing specified actions.
❖	Trajectory: It is a sort of data that records the course of motion or activity taken by a particular object or entity.
❖	Pose: data describing human characteristics, such as head position.
❖	Texture map: Texture maps are used to produce repeating textures, patterns, and distinctive visual effects on the surfaces of 3D models. These can be utilized to define precise aspects such as hair, clothing, and skin to any 3D models.
❖	UV map: A UV map is a flat representation of a 3D model's surface that is used to wrap textures simply. UV unwrapping is the method of creating a UV map. The term U and V relate to the horizontal and vertical axes of the 2D space.

DATA TYPE	DIMENSION	SHAPE INFORMATION	MEMORY PROFICIENCY	COMPUTATION PROFICIENCY	USAGE
RGB	2-D	High	Low	Moderate	Images are detected, represented, and shown in electrical devices like televisions and computers.
Depth	2.5-D	High	Low	Moderate	Simulating the impact of dense semi-transparent material in a scene, such as fog, smoke, or significant amounts of water.
Mesh	3-D	Low	High	Moderate	To form shapes with height, width, and depth, 3D meshes use reference points on the X, Y, and Z axes.
Voxel	3-D	High	Moderate	High	Volumetric imaging in medical and landscape representation in games and simulations.
Point cloud	3-D	Moderate	High	High	from construction and engineering to highway planning and self-driving car development.
Octree	3-D	High	Moderate	Moderate	to recursively subdivide a three-dimensional space into eight octants in order to partition it.
TSDF	3-D	Moderate	High	Moderate	based on a hand-held laser line scanner as a fast, precise, and adaptable geometric fusion method in the 3D reconstruction of industrial products.
Stixel	2.5-D	High	Low	Low	Segmentation, Object tracking.
Texture map	3-D	High	High	High	Generate textures, patterns, or special visual effects.
UV map	3-D	High	Moderate	High	Converting a 3D mesh to a 2D space from a 3D model.

TABLE V: Datasets of facial depth, pose, and recognition

Examples of face images	Dataset	Labelling	Description	camera parameters	APPLICATIONS
	Biwi [75]	3d Position Of The Head And Its Rotation	People Moving Their Heads In Different Directions	Intrinsic + Extrinsic	Automatic Head Pose, Depth, Estimation, Gaze Estimation
	Eure Com Kinectv [76]	Facial Variations, Expressions, Marker Point Positions, Illumination, Occlusion	Performing Various Expressions, Poses	Intrinsic + Extrinsic, Focal Length	Face Recognition, Pose Estimation, Depth Facial Landmark Detection
	3dmda [77]	Spoofing Is Occurring, Eye Positions	3 Different Sessions For All Subjects And Each Session 5 Videos Of 300 Frames Are Captured, Neutral Expression	Intrinsic + Extrinsic	Biometric (Face) Spoofing, Facial Depth Estimation
	Pandora [30]	Head Position And Its Rotation, Features For The Face Verification	People Doing Different Poses In Front Of A Camera Poses	Intrinsic + Extrinsic	Pose, Facial Depth Estimation
	Facescape [78]	Textured 3d Face Models With Pore-Level Geometry, Expressions, Mash, Motion Map, Disparity Map, Texture	Textured 3d Faces, Captured From 938 Subjects And Each With 20 Specific Expressions	Intrinsic + Extrinsic, Focal Length	Predict Elaborate Rig Gable 3d Face Models, Facial Depth Estimation
	Syn Human Face [59]	Expression And Pose, Expressions, Meshes, 3d Position Of The Head And Its Rotation, Lighting	5 Expressions Performed By One Face, Poses, Lighting, Head And Camera Rotation, Translation	Camera Matrix Intrinsic + Extrinsic, Focal Length	Facial Depth Estimation, Pose Estimation
	Baracca Dataset [79]	Measures Of Distance, Age, Weight, Variations, Expressions	In-Car And Outside Views, Human Body Measurements	Intrinsic + Extrinsic	Thermal, Facial Depth Estimation

	Lock3DFace [80]	Changes In Facial Expression, Pose, Occlusion, And Time-Lapse	People Moving Their Heads In Different Directions	Intrinsic + Extrinsic	Pose, Facial Depth Estimation, 3D Face Analysis
	Curtinfaces [81]	Facial Variations, Expressions	Performing Various Expressions, Poses,	Camera Matrix Intrinsic + Extrinsic	Pose, Facial Depth Estimation, Face Recognition
	Iiit-D Rgb-D [82]	Head Position And Its Rotation	Performing Various Expressions Poses,	Camera Matrix Intrinsic + Extrinsic	Face Recognition, Facial Depth Estimation
	Kasparov [46]	Variations, Expressions	Poses, Lighting, Head And Camera Rotation	Intrinsic + Extrinsic	Pose, Facial Depth Estimation

1) BIWI

This dataset [75] comprises 15K images of 20 different subjects which included 6 female subjects and 14 male subjects (4 people were recorded twice). Moreover, this dataset provides the depth image of 640x480 pixels resolution, the corresponding visible image of 640x480 pixels size, and lastly, it also offers the annotation for every image. The depth data is captured using a Kinect v1 sensor. The dataset consist of the head poses with the range of around ± 75 degrees yaw and ± 60 degrees pitch. The overall dataset includes the head's 3D location and rotation as the ground truth data.

2) EURECOM KINECT FACE

This dataset provides multimodal facial data of 52 subjects among which 14 are female, and 38 are male subjects. Eurecom Kinect Face dataset [76] incorporates the depth data which is acquired from Kinect v1 sensor. This data was gathered at different times in the form of two-fold intervals with an average time gap of half month. The recorded data in two different intervals provides the facial frames of each subject in nine situations with various lighting and occlusion conditions and facial expressions which include a neutral face and smiling face.

The provided data incorporates facial data with open mouth, and different occlusions such that strong illumination, eyes occlusion by wearing sunglasses, mouth occlusion by covering it with hand, face side occlusion by placing a paper. The overall dataset provides the RGB colour images, the 3D images, and the depth map which is provided in the forms of the bitmap depth image and the text file containing the actual depth levels acquired from the Kinect sensor. The dataset also incorporates six distinct manual facial landmarks positions which comprise of right and left eye, right and left corner of the mouth, the tip of the nose, and the chin.

3) PANDORA

This dataset [30] provides a total of 250K full-resolution RGB, their corresponding depth data, and their annotations are also included in this dataset. The depth data is acquired from a Kinect v2 sensor. The Pandora dataset is frequently used for

various computer vision tasks such that head poses estimation, head centre localization, and shoulder pose estimation.

4) FACESCAPE

The FaceScape dataset [78] includes large-scale 3D facial models, parametric models, and multi-view images all are recorded in high-quality. The dataset also provides the subject's age and gender, as well as the camera settings configuration. The dataset is made publicly available for non-commercial research purposes. This dataset is consisting of 3D faces acquired from 938 subjects. The overall data comprises 18,760 textured 3D faces, with 20 distinct facial expressions. The dataset provides topological information in all the 3D models by processing pore-level facial geometry. For rough shapes and intricate geometry, fine 3D facial models can be expressed as a 3D morphable model, it is represented as displacement maps. A unique methodology is proposed that takes advantage of the large-scale and high-accuracy dataset by utilizing a deep neural network to extract expression-specific dynamic characteristics.

5) 3DMAD

The 3D Mask Attack Database [77] (3DMAD) contains 76500 frames of 17 different subjects captured using the Kinect v1 depth sensor. Each frame is made up of a depth image with an image dimension of 640x480 pixels – 1x11 bits, a matching RGB image with an image dimension of 640x480 pixels – 3x8 bits, and precisely labelled eye locations (concerning the RGB image). Data is gathered in three distinct sessions for each subject, with each session consisting of five recordings with each recording including 300 frames. The overall data is recorded from the frontal view with neutral expression in controlled environmental conditions. The complete data is gathered in three different sessions. The first two events are for real-world samples, wherein people are recorded for two weeks. A single operator collects 3D mask attacks in the third session (attacker).

6) SYN HUMAN FACE

The SYN Human FACE [59] includes extensive high-quality 3D face models and their corresponding 2D RGB, pixel-accurate ground truth depth images. The suggested framework works as follows: In Character Creator, a collection of virtual human models is built using the real 100 head models. To generate additional data variations, the texture and morphology of the models are modified. These models are then imported to iClone for incorporating the data with five different facial expressions. The mesh, textures, and animation keyframes for the completed iClone models with individual face emotions are then exported in FBX format.

In the next phase head movement (yaw, roll, and pitch) was applied on all the models in Blender to acquire the head pose. The FBX files are then imported and scaled in the Blender world coordinate system. To replicate the real work environment, lights and cameras are included in the scene, whose properties are then adjusted accordingly. The camera sensor near and far clips have been set at 0.01 meters and 5 meters, correspondingly. The sensor size and field of view (FOV) is set to 60 degrees and 36 mm, accordingly. The render layer's RGB and Z-pass outputs are then set up in the compositor to produce the final result. In posture mode, the head and shoulder joints are recognized, the head mesh has pivoted those bones, and the keyframes are stored to apply the rotation.

Finally, the RGB and depth images are created by rendering all of the keyframes. The matching head position (yaw, pitch, and roll) is produced using the Blender software's python module. For each frame, the RGB images are rendered with a resolution size of 640x480 pixels which are then stored in jpg format. Whereas the corresponding depth data is saved in a raw file (.exr format). Moreover, the head poses information for each frame is documented and stored in a text (.txt) file. The rendering process for each 2D frame nearly takes an average time duration of 26.3 seconds which is done using the Cycle Rendering Engine, provided in Blender software which is a type of physically-based path tracer for production rendering. The overall dataset consists of around 3,500k frames, with around 3.5k 2D frames per person.

The data is stored in a separate folder where each folder contains the data of 100 face models. Each face model's produced RGB images, as well as the resulting depth and head posture, are saved in three separate routes for three different backgrounds: plain, textured, and sophisticated. The synthetic dataset was used to create the sample images, which included ground truth depth images and various backdrops (basic, textured, and sophisticated).

7) BARACCA DATASET

The recent interest and growth in depth sensors have supported different methods to instinctively assess the anthropometric measurements, rather than utilising manual procedures and expensive 3D scanners. Normally, the application of depth data is limited due to the lack of depth-based public datasets including accurate anthropometric

annotations. As a result, the authors [79] introduced a better dataset, Baracca, that was constructed specifically for the anthropometric measurements and vehicle perspective, including both in-cabin and outside views. This is a type of multimodal dataset that was created with synchronized depth, infrared, thermal, and RGB cameras to meet the needs of the automobile industry. The depth data is recorded using the Pico Zense DCAM710 depth sensor. The spatial resolution of the RGB sensor is 1920x1080 pixels, whereas the infrared/depth sensor has a resolution of 640x480 pixels. A total of 30 subjects (26 male, and 4 female) took part in the data acquisition process.

8) LOCK3DFACE

The Lock3DFace dataset [80] contains 5671 RGBD facial videos from 509 people, each with a unique facial expression, position, occluded, and moments. The database was collected throughout two periods. The very first event's neutral images are used as training examples, while the final three variations are used to create the 3 test procedures for position, occluded, and expressions. All the images from the second run, in all variants, make up a fourth validation set.

9) CURTINFACES

CurtinFaces [81] is a well-know RGBD face database that includes over 5000 co-registered RGBD images of 52 participants taken using a Microsoft Kinect. The front left, and right postures are the initial three images for each person. The remaining 49 images include 35 images with 5 different illumination variations and 7 different emotions, as well as 7 distinct positions captured with 7 facial variations. Images with sunglasses and arm occluded are also included in this collection.

10) IIIT-D RGB-D

The IIIT-D RGB-D dataset [82] includes 4605 RGBD images from 106 people collected for two periods using a Microsoft Kinect. Each participant was captured with modifications in attitude, emotion, and glasses under typical illumination conditions. The datasets which were before the procedure, which included a 5 cross-validation approach, in the tests set. The head is cropped for each image in the data.

11) KASPAROV

The KASPAROV dataset [46], which comprises automatic facial videos from 108 participants is captured by Microsoft Kinect v1 and v2 cameras. Every subject is shown in videos, each shot at a separate time. A total of 432 videos with 117,831 images are included in the dataset. Because the Kinect v2 sensor data had higher RgbD image registration than the Kinect v1 sensor information.

B. FACIAL DEPTH ESTIMATION LOSS FUNCTIONS

On the reference depth map, deep learning-based algorithms commonly improve a regression model. The key problem for the SoA approaches in deep regression problems is determining a suitable loss function. Neural networks make use of optimization algorithms.

TABLE VI: Publicly Available Depth datasets and properties for faces and poses

DATASET	RGB	DEPTH	VIDEO	POINT-CLOUD	MESH	SCENE	SEMANTIC	OBJECT	CAMERA	ACTION	TRAJECTOR Y	POS E
BIWI [75]	√	√	×	×	×	√	×	×	√	×	×	√
EURECOMKINECT [76]	√	√	×	×	×	√	×	×	√	×	×	√
3DMAD [77]	√	√	×	×	×	√	×	×	√	×	×	×
PANDORA [30]	√	√	×	×	×	√	×	×	√	×	×	×
FACESCAPE [78]	√	√	×	√	√	×	×	×	×	√	√	√
SYN HUMAN FACE [59]	√	√	×	×	√	×	×	√	√	√	×	√
BARACCA DATASET [79]	√	√	×	√	×	×	×	×	√	√	×	√
LOCK3DFACE [80]	√	√	√	×	×	×	×	×	√	√	×	√
CURTINFACES [81]	√	√	×	√	×	×	×	×	√	√	×	√
IIIT-D RGB-D [82]	√	√	×	×	×	×	×	×	√	√	×	√
KASPAROV [46]	√	√	√	×	×	×	×	×	√	√	×	√

No	Dataset Name	Year	Gt	Labeling	Dimension	Objects	Subject/Type	No Images	Diversity	Annotation
1.	BIWI [75]	2011	Depth	Expression, Pose, 2D Skeleton Positions	640 × 480	Multiple	20/realistic	15K	Medium	Real RGB-D
2.	3DMAD [77]	2013	Depth	Expression, Pose, 3D Positions of The Head and its Rotation	640 × 480	Multiple	realistic	76K	Medium	Real RGB-D
3.	CURTINFACES [81]	2013	Depth, Pose	Expression, Pose, 2D Skeleton Positions	640 × 480	Multiple	52/realistic	>5K	High	Real RGB-D
4.	IIIT-D RGB-D [82]	2013	Depth, Pose	Expression, Pose	640 × 480	Multiple	106/realistic	46K	High	Real RGB-D
5.	EURECOM KINECT [76]	2014	Depth	Expression type, Pose, 2D Rotation	256 × 256	Multiple	realistic	20K	Medium	Real RGB-D
6.	LOCK3DFACE [80]	2016	Depth	Expression type, Pose, 3D Position of The Head and Its Rotation	512 × 424	Multiple	509/realistic	>6K	High	Real RGB-D

7.	KASPAROV [46]	2016	Depth	Expression type, Pose, 2D Rotation	64 × 64	Multiple	108/realistic	101K	Medium	Real RGB-D
8.	PANDORA [30]	2017	Depth	Expression, Pose, 2D Skeleton Positions	256 × 256	Multiple	20/realistic	11K	High	Real RGB-D
9.	FACESCAPE [78]	2020	2D, 3D Landmarks, Depth	3D Position of The Head and Its Rotation	4096 × 4096	Multiple	938/Extracted	8K	High	Synthetic, 3D, RGB-B
10.	BARACCA DATASET [79]	2020	Depth	Expression, Pose	640×480	Multiple	30/realistic	>10k	Medium	Real RGB-D
11.	SYN HUMAN FACE [59]	2021	2D, 3D Landmarks, Depth	3D Position of The Head and Its Rotation	640 × 480	Multiple	100/Extracted	350K	High	Synthetic, 3D, RGB-B

This error is calculated using the loss function that evaluates how well or badly the model behaves. Neural depth models have been used to estimate depth from one or many 2-D images using a variety of interesting loss functions for depth estimation challenges. This section lists the common loss functions that are used to estimate facial depth maps from one or multi 2D frame images.

1) ADVERSARIAL LOSS FUNCTION

The binary categorical cross-entropy loss function, which is used for face depth estimation in adversarial training models [20], [21], is defined as follows:

$$L_{bcc}(\mathbf{y}, r) = -\frac{1}{N} \sum_{i=1}^N [r_i \log y_i + (1 - r_i) \log (1 - y_i)] \quad (1)$$

The discriminator output is subjected to $y_i = D(I_i)$, where y_i is the prediction discriminator for the i -th input depth map and r_i is the corresponding ground truth. The goal of the generator model is to create images similar to the GT depth and the discriminator model. The mean squared error (MSE) loss function is used to achieve the first goal.

$$L_{MSE}(y^g, y^d) = \frac{1}{N} \sum_{i=1}^N \|G(y_i^g) - y_i^d\|_2^2 \quad (2)$$

where y^g and y^d are the input images and the output depth map. In the second stage of the network, feed created depth images into the discriminator and use the adversarial loss on the discriminator predictions to see if the generated images can trick the discriminator model. Next, while maintaining the discriminator weights constant, back-propagate the gradients up to the generator model input and modify the generator parameters. As a result, the goal of solving the back-propagation problem is to minimize:

$$\hat{\theta}_g = \arg \min_{\theta_g} L_G(y^g, y^d) \quad (3)$$

Where L_G is a balanced sum of two components and can be defined as:

$$L_G(y^g, y^d) = \lambda \cdot L_{MSE}(y^g, y^d) + L_{bcc}(G(y^g), 1) \quad (4)$$

in which λ is a weighting parameter that controls the influence.

2) GAN LOSS FUNCTION

The loss function [20], [21] in the GAN-based facial depth model is divided into two parts: 1) Generator Loss: The generator loss is the sigmoid cross-entropy loss of the generated images and an array of ones. The L1 loss function (MAE) is utilized to calculate the absolute difference between the target and generated images. This determines how similar the anticipated image is to the actual image. The following formula can be used to compute the total generator loss:

$$L_{Gen_loss} = Gan_loss + \lambda * L1_loss \quad (5)$$

Here λ is set as 100.

$$MAE = \left(\frac{1}{n}\right) \sum_{i=1}^n |r_i - t_i| \quad (6)$$

where r_i is the prediction and t_i are the true value. 2) Discriminator Loss: The discriminator takes real images and generated images as its input. The sigmoid cross-entropy loss of the real images and an array of ones is called real loss. Then the total loss can be calculated by the summation of real loss and the generated loss:

$$T_loss = Real_loss + Generated_loss \quad (7)$$

3) STRUCTURAL SIMILARITY (SSIM) LOSS

SSIM [81] is used to determine the perceived differences between the two similar images. (L_{SSIM}) represents the loss function for the structural similarity index measure (SSIM) and can be defined as:

$$L_{SSIM}(r, t) = \left(\frac{1 - L_{SSIM}(r, t)}{MaxDepth}\right) \quad (8)$$

4) SCALE SHIFT-INVARIANT LOSS

For a single ag image, the scale-shift-invariant loss [81] is defined as

$$L_{SSI}(r, t) = \frac{1}{2N} \sum_i \rho(r, t) \quad (9)$$

where (ρ is the scale-invariant loss).

5) PRE-PIXEL SMOOTHNESS LOSS

Because image gradients commonly have depth inconsistencies, a per-pixel smoothness loss [83] is used in conjunction with the L_{SL} reprojection loss to make the inverse depth prediction better. The following formula is used to determine the (L_{SL}) loss:

$$L_{SL}(r, t) = \sum_i^N \partial_x dt e^{-\partial_x(r,t)} + \partial_y dt e^{-\partial_y(r,t)} \quad (10)$$

Where N denotes the number of valid pixels, ∂d denotes the disparity gradient, and $e^{-\partial_{x,y}(r,t)}$ denotes the edges.

6) RECONSTRUCTION LOSS

When training, the network estimates disparity, and the input image is generated using the bilinear samples, utilized to recreate the image. At the local level, the bilinear sampler is completely differentiable and easily integrated into a network. A L_{Huber} and SSIM is represented as follows: which computes the inconsistencies between both the input image and the regenerated image when coupled as a photometric image reconstruction loss [19].

$$L_R(r, t) = \frac{1}{N} \sum_i^N \frac{1-L_{SSIM}(r,t)}{2} + (1-\alpha)L_{Huber}((r, t)) \quad (11)$$

7) SCALE-INVARIANT LOSS

When training the model, depth estimation methods use the GT depth y and the predicted log depth maps. Scale-invariant loss function [81] (L_{SI}) can be represented by (L_{SI}) for the depth values and is defined as:

$$L_{SI}(r, t) = \frac{1}{N} \sum_i^N (\log(r_i) - \log(t_i))^2 - \frac{\lambda}{N} (\sum_i^N \log(r_i) - \log(t_i))^2 \quad (12)$$

Where λ refers to the balance factor.

8) BERHU LOSS

The OLS estimator is effective in the circumstance of checking for data with outliers or massive errors. Berhu loss, on the other hand, is designed to preserve good attributes in the face of Gaussian noise. Berhu loss function [81] (L_{Berhu}) is defined as:

$$L_{Berhu}(r, t) = \begin{cases} (r_i - t_i) & \text{if } (r_i - t_i) \leq c, \\ \frac{(r_i - t_i)^2 + c^2}{2c} & \text{if } (r_i - t_i) > c, \end{cases} \quad (13)$$

Where r_i, t_i are ground truth and predicted depth maps.

9) HUBER LOSS

MSE is thought to be better at detecting outliers in a dataset, but MAE is expected to be better at preventing them. Data that appear to be outliers, on the other hand, should not be studied, and those points must not be assigned much weight. As a result, the Huber loss function [81] (L_{Huber}) is defined as:

$$L_{Huber}(r, t) = \begin{cases} (r_i - t_i) & \text{if } (r_i - t_i) \geq c, \\ \frac{(r_i - t_i)^2 + c^2}{2c} & \text{if } (r_i - t_i) < c, \end{cases} \quad (14)$$

Where r_i, t_i are ground truth and predicted depth maps.

Table 7 shows the loss function categorized according to their use in depth estimation and their respective use case applications.

TABLE VII: Loss functions categorized in terms of the use case applications.

Loss Function	Purpose Of Usage in Terms of Depth Estimation	Other Use Cases
Adversarial Loss Function [20], [21]	The matching feature vectors of distinct identities are linked together to expand the discriminative characteristics between them. The goal is to change the distance between two facial depth image feature vectors and predict the final depth maps.	Segmentation, 3D reconstruction, Synthetic Data generation
Gan Loss Function [22], [23]	This loss function can be used to penalize inter-subject similarities to force the estimated depth image to preserve as much subject discriminative information as feasible.	Segmentation, 3D reconstruction, Synthetic Data generation
Structural Similarity (SSIM) Loss [81]	<ul style="list-style-type: none"> ✓ The (Structural Similarity Index) loss function is used with the BerHu loss function to use the input image structure and associated features. ✓ The perceptual difference between two similar images is measured by the SSIM loss. Details about structural loss come from relatively adjacent pixels with a deeper connection. ✓ These pixels contain vital information about the structure of the visual scene's objects. 	Classification, Regression, Segmentation
Scale Shift-Invariant Loss [39]	<ul style="list-style-type: none"> ✓ The loss function with the extra term would create a considerably smaller error because the major issue is to preserve relative depth relationships between pixels. ✓ It can also help in a diverse scene such as unknown and inconsistent scales and baselines dataset compatibility. This will allow for data to be trained on from a variety of sensing modalities, including stereo cameras (with potentially unknown calibration), laser scanners, and structured light sensors. 	Regression, Segmentation, Stereo Depth Maps

Pre-Pixel Smoothness Loss	<ul style="list-style-type: none"> ✓ This loss function estimates the similarity between the actual and predicted depth map. ✓ It also benefits the estimated depth-perceptual map's quality. 	Regression, Segmentation, Stereo Depth Maps
Reconstruction Loss [19]	This loss function can be used to make the projected left-view disparity map equal to the projected right-view disparity map, resulting in more realistic disparity maps.	Segmentation, 3D reconstruction, Synthetic Data generation
Scale-Invariant Loss [39]	<ul style="list-style-type: none"> ✓ Regardless of the absolute global size, scale-invariant loss helps in the measurement of relationships between points in the scene. ✓ The average deviation between each pixel depth prediction and the ground truth depth is all that is measured. 	Regression, Segmentation, Stereo Depth Maps
Berhu Loss [40]	<ul style="list-style-type: none"> ✓ BerHu Loss has an advantage since it uses MSE (or L2) loss to give pixels with greater residuals more weight. At the same time, it allows smaller residuals to have a larger effect on gradients than MAE loss. ✓ BerHu's loss function simply combines MAE and MSE, enhancing the whole training process and resulting in more smooth and accurate depth predictions. 	Regression, Segmentation, Stereo Depth Maps
Huber Loss [40]	<ul style="list-style-type: none"> ✓ By balancing the MSE and MAE together, the Huber Loss provides the best of both worlds. ✓ It is less sensitive to outliers in data and can predict more accurate depth maps. 	Regression, Segmentation, Stereo Depth Maps

IV. IMPLEMENTATION DETAILS OF NEURAL DEPTH ESTIMATION NETWORKS

Convolutional neural networks (CNN) are the form of a learning algorithm for data processing with a uniform grid, such as images, that is intended to acquire provides scalable features from low- to high-level structures efficiently and adaptively. Convolution, pooling, and fully connected layers are the three types of layers (or building blocks) that make up CNNs. Convolution and pooling layers are the initial layers that extract features, while the third, a fully connected layer, transmits these characteristics into the final output, such as classification or multiple regression analysis. A convolution layer is an important part of CNN, which is made up of a stack of mathematical computations like

convolution, which is a specific sort of linear operation. Because a feature can appear everywhere in a digital image, image pixels are saved in a two-dimensional (2D) grid, i.e., an array of numbers and a small grid of parameters called the kernel, and an optimizable feature extractor, is implemented at every image position, CNNs are extremely efficient for image analysis. Features extracted can evolve hierarchical structures and progressively more complicated as one layer passes its results into the next layer. Training is the process of adjusting parameters such as kernels to reduce the disparity between outputs and ground truth labels using optimization algorithms like backpropagation and gradient descent. Fig. 2 illustrates the comprehensive implementation details.

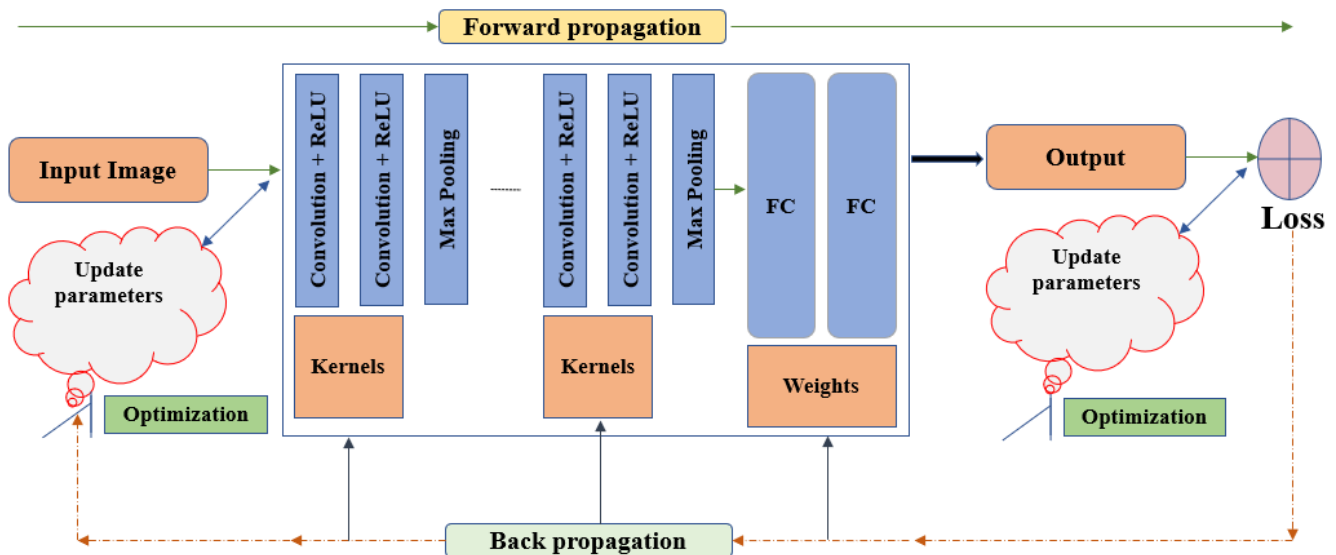


FIGURE 2. A look at the design of a CNN and how it's trained for facial depth estimation. Convolution layers, pooling layers (e.g., max-pooling), and fully connected (FC) layers are the building components that make up a CNN. The success of a model with certain kernels and weights is evaluated using a loss function and forward propagation on a training dataset, and learning parameters, such as kernels and weights, are adjusted using the gradient descent process. The term "corrected linear unit" refers to a linear unit that has been rectified.

The performance of 2D facial depth estimation has been greatly enhanced because of the use of Deep Learning CNNs. Facial depth maps are learned directly from 2D RGB-D facial images by training deep neural networks on large

datasets. Different deep learning models (i.e; VGG, Autoencoder, ResNet, encoder-decoder, inception, DenseNet) are used for facial depth maps which are trained on 2D face depth images. These models typically consist of

CNN, FC, SoftMax layers followed by an appropriate loss function that can minimize the errors of the training networks. Weights of the networks are mostly randomly initialized. The datasets can be augmented in several ways (pose augmentation, resolution, transformation, rotation, cropping, and flipping) using a range of images to enlarge training datasets and can achieve better accuracy. Table 8, shows some comparison analysis of the deep learning-based models for facial depth estimation on iit-d rgb-d [82], kasparov [46], curtin faces [81] and lock3dface [80] datasets. Note that we were unable to compare other qualitative evaluation metrics mentioned in Table 8 due to technical difficulties with publicly available codes and a lack of

instructions for these methods listed in Table 8, and the accuracy results are obtained from their related articles. A CNN-based system has three major components, a training phase, data pre-processing, and model design. To train the model, deep learning-based techniques usually require a significant number of datasets. In CNN-based facial depth maps research, a shortage of large-scale realistic face depth datasets remains an outstanding topic. Because CNN has a lower tolerance for pose changes, suitable data preparation or synthetic data can enhance accuracy before transmitting the data to the model. In addition, selecting an appropriate CNN and loss function are critical.

TABLE VIII: Performance Evaluation of Monocular Depth Estimation based deep learning models on IIIT-D RGB-D [82], KASPAROV [46], CURTIN FACES [81] and LOCK3DFACE [80].

REFERENCE	YEAR	NETWORK	DATASETS	PARAMETERS	LAYERS	INPUT/OUTPUT	ACCURACY %
[46]	2016	AUTOENCODER	IIIT-D RGB-D [82]	47M	CNN, FC, SOFTMAX	RGB/DEPTH	98.7
[84]	2014	VGG-16	KASPAROV [46]	32M	CNN, FC, SOFTMAX	RGB/DEPTH	94.4
[85]	2016	RESNET-50	IIIT-D RGB-D [82]	68M	CNN, FC, SOFTMAX	RGB/DEPTH	95.8
[86]	2017	SE-RESNET-50	CURTIN FACES [81]	86M	CNN, FC, SOFTMAX	RGB/DEPTH	97.8
[58]	2018	INCEPTION-V2	LOCK3DFACE [80]	73M	CNN, FC, SOFTMAX	RGB/DEPTH	71.7
[47]	2020	VGG + DEPTH	IIIT-D RGB-D [82]	84M	CNN, FC, SOFTMAX	RGB/DEPTH	99.6

V. EVALUATION METRICS FOR FACIAL DEPTH ESTIMATION

The most used quantitative metrics for evaluating the performance of monocular facial depth estimation methods are provided in Table 9. These are not limited to 8 metrics, however, most of the published articles used these quantitative metrics to analyze the performance of the trained depth estimation models.

TABLE IX: Quantitative Metrics used for performance evaluation of Monocular Facial Depth Estimation

S.No	Quantitative Metrics Name	Formula
------	---------------------------	---------

1	AbsRel	$\frac{1}{N} \sum \frac{ d_i - d_i^* }{d_i}$
2	RMSE	$\sqrt{\frac{1}{N} \sum d_i - d_i^* ^2}$
3	RMSE (log)	$\sqrt{\frac{1}{N} \sum \log d_i - \log d_i^* ^2}$
4	SqRel	$\frac{1}{N} \sum \frac{ d_i - d_i^* ^2}{d_i}$
5	Accuracies	% of $d_i \max(d_i/g_i) = \delta thr$
6	L1	$\sum_{i=1}^n y_{true} - y_{predicted} $
7	L2	$\sum_{i=1}^n (y_{true} - y_{predicted})^2$

8	NRMSE	$\sqrt{\frac{1}{N} \sum \frac{ d_i - d_i^* ^2}{d_i^2}}$
where d_i and d_i^* are the ground truth and predicted depth at pixel i and N is the total number of pixels.		

VI. FACIAL DEPTH ESTIMATION MODEL

Many consumer applications including robotics, augmented reality and advanced driving monitoring systems can benefit from facial depth estimation neural depth networks from single images. A methodology for creating depth maps from single images of human faces is presented in this section, which utilizes the source face depth and corresponding ground truth depth using neural networks.

Existing facial depth map algorithms may produce depth maps with comparable accuracy, but they suffer from difficulties such as missing values and depth similarities, which result in holes in depth images. As an alternative, the model used in this study automates the collection of optimal parameters, reducing model complexity during the training process for facial depth estimation.

A recent SoA LapDepth [68] model is chosen to accomplish high-quality facial depth estimation from a single 2D frame. By applying the Laplacian pyramid-based decomposition technique to the decoding process, the suggested method intends to successfully restore local details (i.e., depth boundaries) as well as the global layout of the depth map. The depth residual including local details, which suitably describe depth attributes of different scale-spaces, is created using Laplacian residuals of the input colour image guidance encoded features. To improve the efficiency of this decoding

process, the authors [87] introduce weight standardization to the pre-activation convolution block, which greatly helps in estimating depth residuals. First, describe the overall architecture of the proposed decoder for monocular facial depth estimation in this section. The entire decoding procedure will then be detailed, including the influence of weight standardization. Finally, the loss functions utilized to train the model architecture are discussed.

A. ARCHITECTURE DETAILS

The proposed neural depth network for single image facial depth maps mechanism is provided in this section, as well as the suggested loss function for improving the training process over the training data.

1) ENCODER MODEL

The proposed method's general architecture is demonstrated in Fig. 3. [87]. The suggested decoder for restoring depth residuals is connected to the pre-trained encoder in the network. ResNext10 [56] is used in the encoder phase, which has been pre-trained for image classification. The input colour image is compressed as latent information using densely layered convolution blocks on the encoder. The spatial size of such features shrinks to a fraction of the original resolution, but they compactly contain the colour-depth relationship in the embedding space, which is learned from various scene geometries. For the convolution block of the encoder, the authors utilize the Dense ASPP approach [88] with four dilation rates of 3, 6, 12, and 18 to extract more dense contextual information.

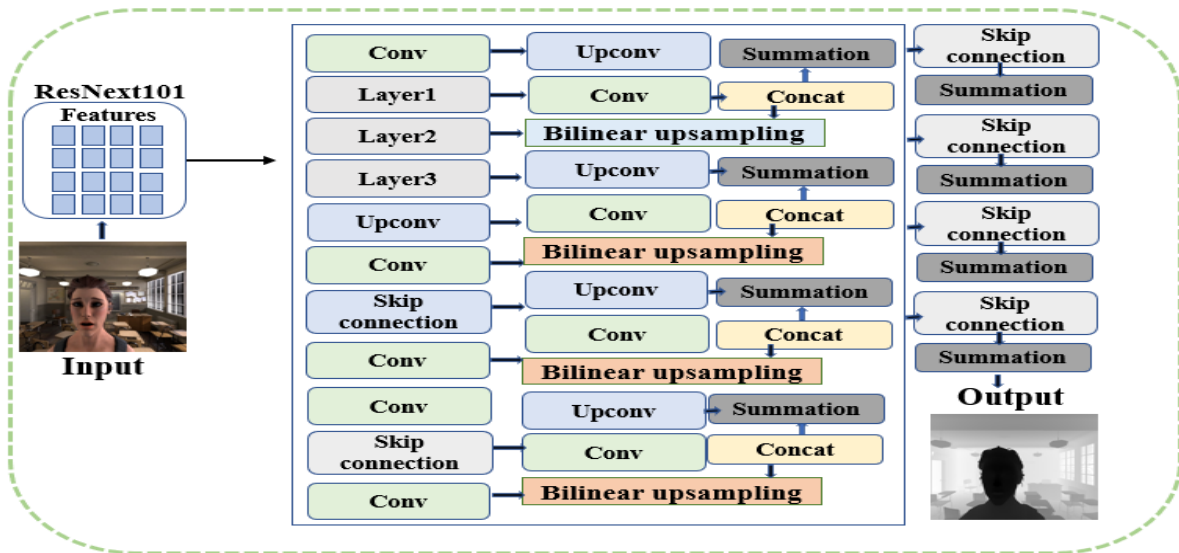


FIGURE 3. The overall architecture of the proposed method for monocular facial depth estimation.

The suggested decoder is separated into many Laplacian pyramid branches. One branch, which is in charge of the Laplacian pyramid's topmost level, undertakes decoding work to restore the depth map's global layout. The depth residuals are generated by other branches using latent features led by Laplacian residuals of the input colour image

at the matching scale. Using point-wise addition, this depth residual is gradually integrated with the middle depth map, which is the result of the higher level of the Laplacian pyramid. The decoding technique is based on a five-level Laplacian pyramid. All convolution layers in the decoder have a filter size of 3x3.

2) DECODER MODEL

The laplacian residual of the input colour image is derived in the first phase. For all scaling methods in the suggested methodology, downsampling the initial input image, upsampling, and bilinear interpolation are used. Concatenated features are input into layered convolution blocks, and the output is added pixel-by-pixel. The one-channel output, which is made up of stacked convolution blocks, has the same spatial resolution as the input colour image. It's important to note that input guides the decoding process to precisely restore local characteristics of various size areas, which aids in revealing depth boundaries without distortions. Finally, starting at the top of the Laplacian pyramid, the depth map is gradually recreated. The weight standardization in the pre-activation convolution block, which is the core module of the decoder, is made to produce the decoding process for monocular facial depth estimation more effectively. Because the depth map is reconstructed using an iterative accumulation of depth residuals, it is preferable for the projected depth residual to have a balancing of negative and positive values to estimate depth information reliably and accurately. During backpropagation, which is calculated from each layer of the laplacian pyramid, the decoder is capable of improving the flow of gradient by normalizing them. This is preferable for maintaining the colour-to-depth translation's stability based on residual information. The procedure is anticipated to be able to effectively understand the important connection between colour and depth values for facial images by combining this benefit with the Laplacian pyramid-based decomposition technique.

B. LOSS FUNCTION

The facial depth estimation task's final goal is to find a function that predicts the depth from an input image. (L_{silog}) is the most common loss function that is found in the literature more helpful for depth estimation, The network's trainable parameters are tuned based on the loss function, which employs properly scaling the loss function's range can improve converging and training outputs while putting a stronger focus λ on decreasing error variance, leading in a Silog loss function. [89] (L_{silog}) is defined:

$$L_{si}(d_i, d_i^*) = \frac{1}{N} \sum_i^N (\log(d_i) - \log(d_i^*))^2 - \frac{\lambda}{N} (\sum_i^N \log(d_i) - \log(d_i^*))^2 \quad (15)$$

where λ is the balance factor and N is the number of pixels.

By rewriting the equation. 15:

$$L_{silog}(d_i, d_i^*) = \frac{1}{N} \sum_i^N (\log(d_i) - \log(d_i^*)) - \frac{1}{N} \sum_i^i (d_i - d_i^*)^2 + (1 - \lambda) \frac{1}{N} \sum_N^i (d_i - d_i^*)^2 \quad (16)$$

In log space, the combined Silog loss is defined as:

$$L_{silog}(d_i, d_i^*) = \alpha \sqrt{L_{silog}(d_i, d_i^*)} \quad (17)$$

VII. EXPERIMENTAL RESULTS

The experimental results are presented in this section show how well the proposed model performs. The purpose of these experiments is to see how well synthetic facial depth data can be used to estimate facial depth estimation. A set of SoA depth estimation single image neural networks is used to analyze and compare the human facial depth estimation. Furthermore, the model is first trained on a synthetic human facial depth dataset, after which it is evaluated against four different datasets (Pandora, Eurecom Kinect Face, Biwi Kinect Head Pose, and Synthetic human face datasets) explained in section 3. After that, there is a brief comparison analysis (evaluation results of the SoA to the proposed model) is presented. The experiments show that a model trained on a large and diverse set of facial depth images, along with the appropriate training methods, produce SoA results in a variety of scenarios. The zero-shot cross-dataset transfer technique is used to demonstrate this process.

A. TRAINING METHODOLOGY

The proposed approach is designed in the PyTorch tool. The suggested decoder's parameters (i.e., the network's weights) are all initialized using the approach described in [88]. The proposed decoder has group normalization in each layer, which is known to be batch size independent. The model is trained on a synthetic human facial depth dataset (described in section 3), which was divided into training and validation sets with 0.8 and 0.2 ratios for facial depth estimation. The network is trained using the Adam optimizer for 50 epochs with a batch size of 6, with power and momentum set to 0.9 and 0.999, respectively. For the encoder and decoder, the weight decaying factor is set to 0.0005 and 0. Using a polynomial decay with the power of 0.5, the learning rate is first set to 10^{-4} and then gradually decreased until it reaches 10^{-5} . The overall training process is conducted on a machine equipped with two TITAN 1080 GPUs, which takes a time duration of 72 hours. The model has 73M parameters and to avoid overfitting, the online data augmentation method is used in the training process. For the SYN HUMAN FACE dataset, training samples are randomly cropped to 512x416 pixels before being randomly rotated in the range of [3, 3] degrees. With a ratio of 0.5, input images are also horizontally flipped. Furthermore, the scale factor picked from the range of [0.9, 1.1] is used to alter the brightness, colour, and gamma values of the input colour images.

B. EXPERIMENTAL DETAILS AND RESULTS

The first phase of this subsection explains the training dataset that was used to train the neural depth model for facial depth estimation. The second part explains the testing and evaluation process used to evaluate the model's generalization performance. 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 defined in

Table 9. Four test datasets were chosen based on the diversity and accuracy of their ground truth. The model's performance is compared to existing SoA approaches in the final phase. Table 10 summarizes all of the information from this study's experiments.

TABLE X: Information about how experiments have been conducted

Method	LapDepth [87]
Tools/Software	PyTorch, Open3d
Training Time	72 hours
Input	512×416
Output	512×416
Type	CNN (Encoder-Decoder)
Optimizer	Adam
Learning Rate	10 ⁻⁵
Environment	2×TITAN 1080 GPUs 2.5Ghz Python
Memory	16×2GB
Epochs	50
Parameters	73M

1) MODEL TRAINING DATASET

The synthetic human facial dataset having various variations including camera location, light position, body-pose, facial animations, scene illuminations, and pixel-accurate ground truth depth is used for training the proposed neural depth model for facial depth maps. This dataset is briefly explained in (section 3-part A subsection 6. Before conducting any experiments, the training data is processed and split into

three sets: training set 80%, validation set 20%, and test set 10%, each having its ground truth depth.

2) TEST DATASETS

For comparison purposes, the zero-shot cross-dataset transfer protocol is utilized. The model was trained on a single dataset before being tested on unseen test datasets. The four datasets described in (section 3-part A) were chosen for testing and evaluation (i.e, Pandora, Eurecom Kinect Face, Biwi Kinect Head Pose, and Synthetic human face datasets).

3. MODEL PERFORMANCE EVALUATION

The performance of the facial depth estimation model LapDepth [87] is compared to the SoA models (i.e; **MiDaS** [90], **DPT** [91] and **BTS** [89]) on the synthetic human facial dataset in Fig. 4 and Table 11. All of the training and testing experiments in this work have been coded and are available on Github. The network achieves SoA results, as shown in Table 11. The proposed model qualitative results against SoA approaches are shown in Fig. 5 and Fig. 6. As shown in Fig. 5, the results demonstrated a details information and consistency, indicating that the proposed chosen approach works better at facial depth estimation. The model outperformed SoA both numerically and qualitatively in tests across a variety of real and synthetic images and set a new SoA for facial depth estimation.

In comparison to other SoA methods, the LapDepth approach performed best in terms of accuracy and depth range, according to the comparison analysis Table 11 and Fig. 6. As shown in Table 11, the network achieved 0.0281 RMSE and 0.9976 threshold accuracy on a synthetic human facial dataset (row 8). For better visualization, the results are shown in the different colour maps. Note that, predicted depth images (Greys) indicate the inverse depth map Fig 4.

TABLE XI: QUANTITATIVE EVALUATIONS ON THE SNY HUMAN FACE DATASET[59]

No.	Methods	AbsRel	SqRel	RMSE	RMSElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
1.	DenseDepth-169 [92]	0.0296	0.0096	0.0373	0.0129	0.9890	0.9920	0.9981
2.	ResNet-101 [59]	0.0123	0.0210	0.0306	0.0089	0.9938	0.9965	0.9980
3.	EfficientNet-B0 [93]	0.0145	0.0280	0.0360	0.0154	0.9912	0.9934	0.9978
4.	BTS [89]	0.0165	0.0092	0.0206	0.0102	0.9830	0.9943	0.9956
5.	UNet-simple [94]	0.0103	0.0207	0.0281	0.0089	0.9960	0.9976	0.9987
6.	MiDaS [90]	0.0146	0.0204	0.03560	0.0323	0.9665	0.9902	0.9956
7.	DPT [91]	0.0156	0.0106	0.0394	0.0184	0.9567	0.9646	0.9943
8.	LapDepth [87]	0.0145	0.0041	0.0204	0.3614	0.9545	0.9857	0.99582

As mentioned before the most commonly used quantitative metrics for evaluating the performance of trained monocular facial depth estimation methods are provided in Table 9.

Based on the metrics in Table 11 i.e.; RMSE, RMSElog, SqRel, AbsRel, and accuracies one can compare and decide which method performance is better.



FIGURE 4. Qualitative results in a sample of the synthetic human facial test dataset that was not used for training or validation. Input RGB images, ground truth images, predicted depth images, predicted depth images (Greys), and predicted depth images are shown from left to right.

The model is compared with the SoA models (i.e; **MiDaS** [90], **DPT** [91], and **BTS** [89]) for comparison, and the qualitative results are shown in Fig. 5. We were unable to train the techniques (i.e. MiDaS, DPT) from scratch due to unavailability of the training codes and a lack of instructions, and hence simply fine-tuned the model checkpoint for testing and validation purposes. The method BTS is initially trained on a training dataset before being put to the test on four different datasets. The suggested method has an advantage over the BTS and other SoA methods, as shown in Fig. 5. The model can recover fine details such as facial information and backgrounds since it is trained on pixel-accurate ground truth depth facial data. Pandora, Eurecom Kinect Face, and Biwi Kinect Head Pose are among the datasets that rarely capture those details. It is difficult to learn when training

neural depth networks due to a very sparse ground truth depth. It is noticed that the method LapDepth successfully preserves the facial depth information even with complicated geometries as compared to the rest of the SoA approaches. As can be seen in Fig. 6, the results show improved information and consistency, demonstrating that the works were better at depth estimation on real facial depth datasets. The network was not used for training or validation, and the method was exclusively trained on synthetic human facial depth datasets and tested on real datasets. In fig. 5, the results in the 4th column predicted depth images (Greys) indicate the inverse depth maps that is originally used by MiDaS [90]. The rest of the comparison results are respectively calculated with the same scale while predicting the depth estimation models.



FIGURE 5. From left to right, qualitative results of facial monocular depth estimation algorithms (Input: input RGB images; GT: ground truth images; Ours: LapDepth [87], MiDaS [90], DPT [91], and BTS [89] applied to the Synthetic human facial dataset [59]).

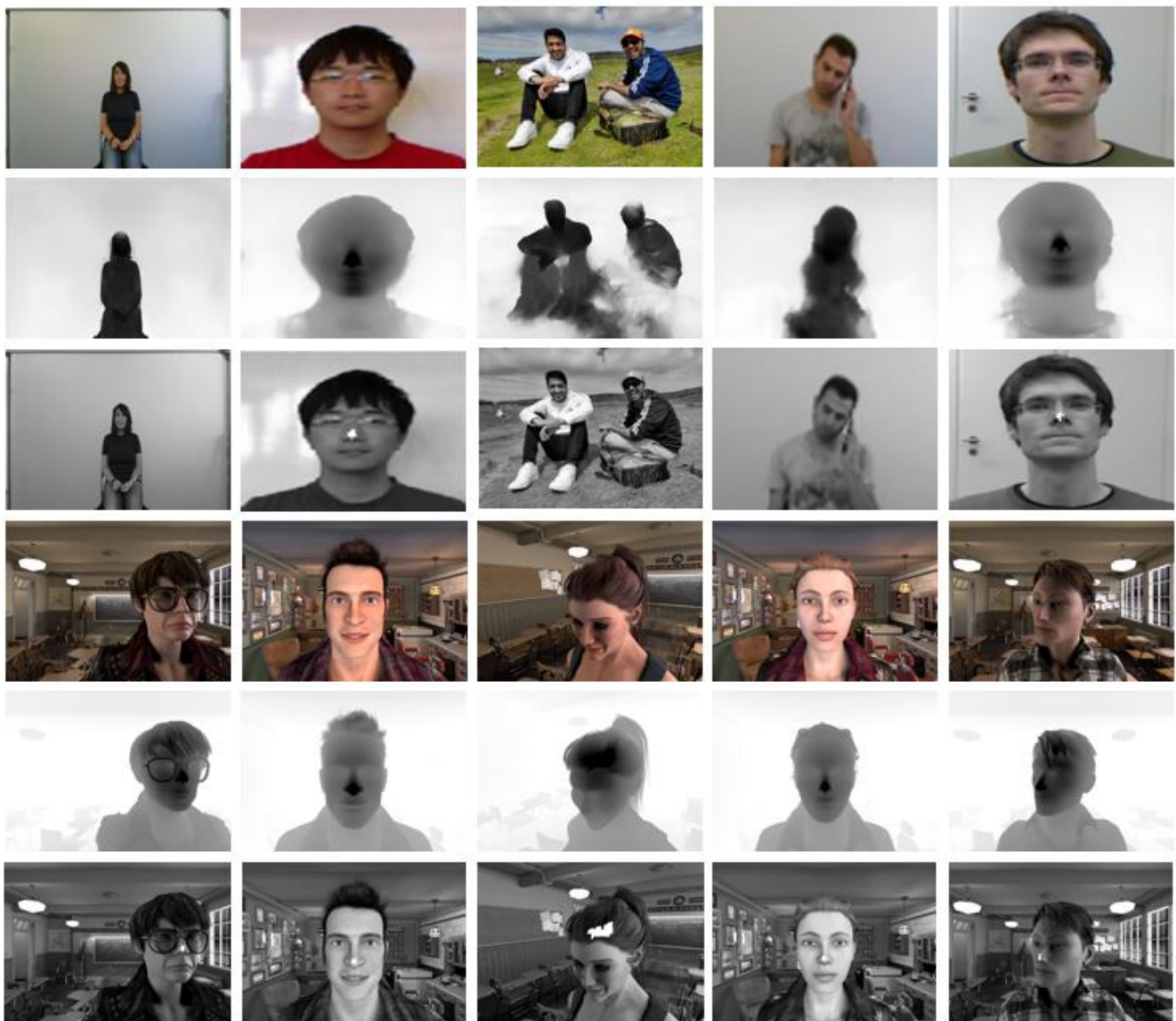


FIGURE 6. The results of a facial monocular depth estimation method's qualitative evaluation. It demonstrates how to use data from several, independent sources to estimate facial depth in a single view, despite changing and unknown depth range and scale. The method allows for broad generalization across datasets. Input images at the top. Middle: depth maps predicted by the approach provided. Bottom: corresponding point clouds as seen from a different perspective. Open3D [95] was used to render point clouds. Images from the Synthetic human facial dataset, the Pandora dataset, the Eurecom Kinect Face dataset, and the Biwi Kinect Head Pose dataset, as well as a real image of the main authors that were not seen during training.

VIII. DISCUSSION

The results presented in the previous section are discussed in the following section.

1. The model is trained by using only the Synthetic Human Facial Depth Dataset and evaluated against four different datasets, including the Pandora dataset, Eurecom Kinect Face dataset, Biwi Kinect Head Pose dataset, and the test Synthetic Human Facial Depth Dataset, as well as real images, in the testing phase. The results demonstrate that the trained model outperforms the other SoA approaches MiDaS, DPT, and BTS. It is important to mention that the low size and diversity of the Pandora dataset, Eurecom Kinect Face dataset, Biwi

Kinect Head Pose dataset do not perform well on the generalization performance of the studied models, as shown in Fig. 6. Furthermore, most depth GT are error-prone due to practical restrictions in data gathering. The depth GT data is particularly prone to mistakes in these datasets that make it difficult for models to learn robust facial depth information.

2. Synthetic facial data will, of course, lack the same level of detail in terms of skin features as compared to real-world image data. However, considering the numerous advantages of utilizing synthetic data to train a neural depth model, it acquires comparable accuracy to real-world data as shown in Fig. 6.

3. When the new loss function is utilized in the final set of experiments, the model outperforms SoA when the network is trained entirely on synthetic data. As a result, it is rational to assume that employing a scalable loss function and training technique helps in acquiring greater accuracy and facial depth information.
4. The model measure how effectively the created faces preserve the individual visual features of the subjects, which requires both high and low-level features to work effectively. The suggested model allows for the maximum test accuracy and outperforms the previous models that have been examined. Based on the results, the model can estimate both high-level and low-level aspects of facial depth maps, resulting in realistic and discriminative results.
5. Using the model predicted depth maps, as shown in Fig. 6 (row 3 and 6), the corresponding point clouds can be generated from a different perspective. Many developing visual applications require quick, direct, and exact depth information, which points clouds deliver. To localize and navigate, autonomous technologies such as robots, augmented reality devices, and self-driving cars rely on depth. In high-end smartphones, depth also enables computational photography functions like auto focus and portrait mode, which are especially useful at night when depth is difficult to obtain with traditional cameras but is readily available from a LiDAR.

IX. CONCLUSION AND FUTURE RESEARCH

This paper investigated the comprehensive details of facial depth datasets and loss functions generated in the field of computer vision for facial depth estimation problems. In various facial depth map tasks based on deep learning networks, publicly available facial depth datasets and facial depth-based loss functions have obtained robust results. The facial depth datasets are utilized in a variety of applications, including person detection and action recognition, face and pose detection, and biomedical applications. Implementation details of how neural depth networks work, as well as their associated evaluation matrices, are presented in this study. In addition to this, SoA neural architecture for facial depth estimation is proposed, along with a comparison evaluation. The proposed model outperforms current SoA techniques when tested against four different datasets. The proposed method's unique loss function helps the network in learning information aspects more robustly thus providing a detailed prediction. The training is done using synthetic human facial depth datasets, while the evaluation is done with real as well as synthetic facial images. The results prove that the proposed neural model outperforms current SoA networks, thus establishing a new benchmark for facial depth mapping and research aspects. Also, the achieved results presented in this paper can be utilized as a reference for better facial depth estimation model design and validation purposes.

Future research can be focused on developing more robust neural networks, as well as paying more attention to the newly developed facial depth datasets to obtain pixel-accurate ground truth depth maps. Because the currently available datasets have issues, particularly with realistic human faces, they can be employed in a range of real-world applications such as in-cabin driver monitoring, robotics, and 3D face reconstructions if these difficulties are addressed.

Finally, the available SoA depth estimation models can be reconsidered for the prediction of facial depth maps because they are mostly used for indoor and outdoor scene tasks and have not been extensively studied for human faces. They can also be investigated for other tasks such as single view facial recognition and surface normal prediction, 3D reconstructions, and while training on datasets both real and synthetic. The GitHub code is available online and can be found at this URL

<https://github.com/khan9048/LapDepth-for-Facial-depth-estimation->

REFERENCES

- [1] S.-F. Wang and S.-H. Lai, "Reconstructing 3D face model with associated expression deformation from a single face image via constructing a low-dimensional expression deformation manifold," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 2115–2121, 2011.
- [2] L. Spreewers, "Fast and accurate 3d face recognition," *Int. J. Comput. Vis.*, vol. 93, no. 3, pp. 389–414, 2011.
- [3] R. Lengagne, P. Fua, and O. Monga, "3D stereo reconstruction of human faces driven by differential constraints," *Image Vis. Comput.*, vol. 18, no. 4, pp. 337–343, 2000.
- [4] J. Choi *et al.*, "3D face reconstruction using a single or multiple views," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 3959–3962.
- [5] C. K. Chow and S. Y. Yuen, "Recovering shape by shading and stereo under lambertian shading model," *Int. J. Comput. Vis.*, vol. 85, no. 1, pp. 58–100, 2009.
- [6] H.-S. Koo and K.-M. Lam, "Recovering the 3D shape and poses of face images based on the similarity transform," *Pattern Recognit. Lett.*, vol. 29, no. 6, pp. 712–723, 2008.
- [7] Z.-L. Sun, K.-M. Lam, and Q.-W. Gao, "Depth estimation of face images using the nonlinear least-squares model," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 17–30, 2012.
- [8] J. Fortuna and A. M. Martinez, "Rigid structure from motion from a blind source separation perspective," *Int. J. Comput. Vis.*, vol. 88, no. 3, pp. 404–424, 2010.
- [9] Z.-L. Sun and K.-M. Lam, "Depth estimation of face images based on the constrained ICA model," *IEEE Trans. Inf. Forensics Secur.*, vol. 6, no. 2, pp. 360–370, 2011.
- [10] K. Konda and R. Memisevic, "Unsupervised learning of depth and motion," *arXiv Prepr. arXiv1312.3429*, 2013.
- [11] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun, "Continuous markov random fields for robust stereo estimation," in *European Conference on Computer Vision*, 2012, pp. 45–58.
- [12] P. Cavestany, A. L. Rodriguez, H. Martinez-Barbera, and T. P. Breckon, "Improved 3D sparse maps for high-performance SFM with low-cost omnidirectional robots," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 4927–4931.
- [13] L. Ding and G. Sharma, "Fusing structure from motion and lidar for dense accurate depth map estimation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 1283–1287.
- [14] M. W. Tao, P. P. Srinivasan, J. Malik, S. Rusinkiewicz, and R. Ramamoorthi, "Depth from shading, defocus, and correspondence using light-field angular coherence," in

- Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1940–1948.
- [15] R. J. Woodham, “Photometric method for determining surface orientation from multiple images,” *Opt. Eng.*, vol. 19, no. 1, p. 191139, 1980.
- [16] A. Atapour-Abarghouei and T. P. Breckon, “Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2800–2810.
- [17] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” *Adv. Neural Inf. Process. Syst.*, vol. 3, no. January, pp. 2366–2374, 2014.
- [18] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, “Unsupervised cnn for single view depth estimation: Geometry to the rescue,” in *European conference on computer vision*, 2016, pp. 740–756.
- [19] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6602–6611, 2017, doi: 10.1109/CVPR.2017.699.
- [20] A. T. Arslan and E. Seke, “Face depth estimation with conditional generative adversarial networks,” *IEEE Access*, vol. 7, pp. 23222–23231, 2019.
- [21] R. Dovgand and R. Basri, “Statistical symmetric shape from shading for 3D structure recovery of faces,” in *European Conference on Computer Vision*, 2004, pp. 99–113.
- [22] W. A. P. Smith and E. R. Hancock, “Recovering facial shape using a statistical model of surface normal direction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1914–1930, 2006.
- [23] W. Y. Zhao and R. Chellappa, “Symmetric shape-from-shading using self-ratio image,” *Int. J. Comput. Vis.*, vol. 45, no. 1, pp. 55–75, 2001.
- [24] Q. Jin, J. Zhao, and Y. Zhang, “Facial feature extraction with a depth AAM algorithm,” in *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, 2012, pp. 1792–1796.
- [25] C. Jordan, “Feature Extraction from Depth Maps for Object Recognition,” 2013.
- [26] I. Kemelmacher-Shlizerman and R. Basri, “3D face reconstruction from a single image using a single reference face shape,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 394–405, 2010.
- [27] A. T. Baby, A. Andrews, A. Dinesh, A. Joseph, and V. K. Anjusree, “Face Depth Estimation and 3D Reconstruction,” in *2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)*, 2020, pp. 125–132, doi: 10.1109/ACCTHPA49271.2020.9213233.
- [28] J. Zhang, K. Li, Y. Liang, and N. Li, “Learning 3D faces from 2D images via stacked contractive autoencoder,” *Neurocomputing*, vol. 257, pp. 67–78, 2017.
- [29] F. Zhang, N. Liu, Y. Hu, and F. Duan, “MFFNet: Single facial depth map refinement using multi-level feature fusion,” *Signal Process. Image Commun.*, p. 116649, 2022.
- [30] 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.
- [31] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, 2015.
- [32] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional pose machines,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.
- [33] S. Wang, Z. Cheng, X. Deng, L. Chang, F. Duan, and K. Lu, “Leveraging 3D blendshape for facial expression recognition using CNN,” *Sci. China Inf. Sci.*, vol. 63, no. 120114, pp. 1–120114, 2020.
- [34] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, “Deep layer aggregation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2403–2412.
- [35] Q. Yang, R. Yang, J. Davis, and D. Nistér, “Spatial-depth super resolution for range images,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [36] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, “Joint bilateral upsampling,” *ACM Trans. Graph.*, vol. 26, no. 3, pp. 96–es, 2007.
- [37] A. K. Riemens, O. P. Gangwal, B. Barenbrug, and R.-P. Berretty, “Multistep joint bilateral depth upsampling,” in *Visual communications and image processing 2009*, 2009, vol. 7257, pp. 192–203.
- [38] A. Foi, V. Katkovnik, and K. Egiazarian, “Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images,” *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1395–1411, 2007.
- [39] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, “Adaptive deblocking filter,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, 2003.
- [40] V. Katkovnik, K. Egiazarian, and J. Astola, “Local approximation techniques in signal and image processing,” 2006.
- [41] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, 1998, pp. 839–846.
- [42] C. Angermann, M. Schwab, M. Haltmeier, C. Laubichler, and S. Jónsson, “Unsupervised Single-shot Depth Estimation using Perceptual Reconstruction,” *arXiv Prepr. arXiv2201.12170*, 2022.
- [43] A. Sepas-Moghaddam, P. L. Correia, K. Nasrollahi, T. B. Moeslund, and F. Pereira, “Light field based face recognition via a fused deep representation,” in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2018, pp. 1–6.
- [44] L. Jiang, J. Zhang, and B. Deng, “Robust RGB-D face recognition using attribute-aware loss,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2552–2566, 2019.
- [45] G. Mu, D. Huang, G. Hu, J. Sun, and Y. Wang, “Led3D: A lightweight and efficient deep approach to recognizing low-quality 3D faces,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5773–5782.
- [46] A. Chowdhury, S. Ghosh, R. Singh, and M. Vatsa, “RGB-D face recognition via learning-based reconstruction,” in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2016, pp. 1–7.
- [47] H. Zhang, H. Han, J. Cui, S. Shan, and X. Chen, “RGB-D face recognition via deep complementary and common feature learning,” in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, 2018, pp. 8–15.
- [48] S. Pini, G. Borghi, R. Vezzani, D. Maltoni, and R. Cucchiara, “A Systematic Comparison of Depth Map Representations for Face Recognition,” *Sensors*, vol. 21, no. 3, p. 944, 2021.
- [49] V. Le, H. Tang, L. Cao, and T. S. Huang, “Accurate and efficient reconstruction of 3d faces from stereo images,” in *2010 IEEE International Conference on Image Processing*, 2010, pp. 4265–4268.
- [50] Y. Zheng, J. Chang, Z. Zheng, and Z. Wang, “3d face reconstruction from stereo: A model based approach,” in *2007 IEEE International Conference on Image Processing*, 2007, vol. 3, pp. III–65.
- [51] J. R. A. Moniz, C. Beckham, S. Rajotte, S. Honari, and C. Pal, “Unsupervised depth estimation, 3d face rotation and replacement,” *arXiv Prepr. arXiv1803.09202*, 2018.
- [52] M. Abdelrahman, A. Ali, S. Elhabian, H. Rara, and A. A. Farag, “A passive stereo system for 3D human face reconstruction and recognition at a distance,” in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 17–22.
- [53] G. Kanojia and S. Raman, “FacialStereo: Facial depth estimation from a stereo pair,” in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, 2014, vol. 3, pp. 686–691.
- [54] A. Aissaoui, J. Martinet, and C. Djeraba, “Rapid and accurate face depth estimation in passive stereo systems,” *Multimed. Tools Appl.*, vol. 72, no. 3, pp. 2413–2438, 2014.

- [55] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1851–1858.
- [56] M. Reiter, R. Donner, G. Langs, and H. Bischof, *Estimation of face depth maps from color textures using canonical correlation analysis*. na, 2006.
- [57] D. Kong, Y. Yang, Y.-X. Liu, M. Li, and H. Jia, "Effective 3d face depth estimation from a single 2d face image," in *2016 16th International Symposium on Communications and Information Technologies (ISCIT)*, 2016, pp. 221–230.
- [58] J. Cui, H. Zhang, H. Han, S. Shan, and X. Chen, "Improving 2D face recognition via discriminative face depth estimation," in *2018 International Conference on Biometrics (ICB)*, 2018, pp. 140–147.
- [59] 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 Networks*, vol. 142, pp. 479–491, 2021, doi: 10.1016/j.neunet.2021.07.007.
- [60] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color! Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–11, 2016.
- [61] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *European conference on computer vision*, 2016, pp. 577–593.
- [62] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," *arXiv Prepr. arXiv1609.03126*, 2016.
- [63] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv Prepr. arXiv1411.1784*, 2014.
- [64] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 469–477, 2016.
- [65] D. Huang, K. Ouji, M. Ardabilian, Y. Wang, and L. Chen, "3D face recognition based on local shape patterns and sparse representation classifier," in *International Conference on Multimedia Modeling*, 2011, pp. 206–216.
- [66] J. Lee, B. Bhattarai, and T.-K. Kim, "Face Parsing from RGB and Depth Using Cross-Domain Mutual Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1501–1510.
- [67] M. Fabbri, G. Borghi, F. Lanzi, R. Vezzani, S. Calderara, and R. Cucchiara, "Domain translation with conditional gans: from depth to rgb face-to-face," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 1355–1360.
- [68] G. Borghi, M. Fabbri, R. Vezzani, S. Calderara, and R. Cucchiara, "Face-from-depth for head pose estimation on depth images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 596–609, 2018.
- [69] S. Berretti, A. Del Bimbo, and P. Pala, "3D face recognition using isogeodesic stripes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2162–2177, 2010.
- [70] Y. Wang, J. Liu, and X. Tang, "Robust 3D face recognition by local shape difference boosting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1858–1870, 2010.
- [71] T. C. Faltemier, K. W. Bowyer, and P. J. Flynn, "A region ensemble for 3-D face recognition," *IEEE Trans. Inf. Forensics Secur.*, vol. 3, no. 1, pp. 62–73, 2008.
- [72] F. Zhang, N. Liu, L. Chang, F. Duan, and X. Deng, "Edge-guided single facial depth map super-resolution using CNN," *IET Image Process.*, vol. 14, no. 17, pp. 4708–4716, 2021.
- [73] L. Jovanov, A. Pižurica, and W. Philips, "Denoising algorithm for the 3D depth map sequences based on multihypothesis motion estimation," *EURASIP J. Adv. Signal Process.*, vol. 2011, no. 1, pp. 1–17, 2011.
- [74] S. Yang, S. Song, Q. Guo, X. Lu, and J. Liu, "Facial depth map enhancement via neighbor embedding," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2016, pp. 1249–1254.
- [75] G. Fanelli, T. Weise, J. Gall, and L. Van Gool, "Real time head pose estimation from consumer depth cameras," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6835 LNCS, pp. 101–110, 2011, doi: 10.1007/978-3-642-23123-0_11.
- [76] 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, 2014.
- [77] E. Nesli and S. Marcel, "Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect," in *IEEE 6th International Conference on Biometrics: Theory, Applications and Systems (BTAS'13)*, 2013, pp. 1–8.
- [78] H. Yang *et al.*, "Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 601–610.
- [79] S. Pini, A. D'Eusanio, G. Borghi, R. Vezzani, and R. Cucchiara, "Baracca: a multimodal dataset for anthropometric measurements in automotive," in *2020 IEEE International Joint Conference on Biometrics (IJCB)*, 2020, pp. 1–7.
- [80] J. Zhang, D. Huang, Y. Wang, and J. Sun, "Lock3DFace: A large-scale database of low-cost Kinect 3D faces," in *2016 International Conference on Biometrics (ICB)*, 2016, pp. 1–8.
- [81] B. Y. L. Li, A. S. Mian, W. Liu, and A. Krishna, "Using Kinect for face recognition under varying poses, expressions, illumination and disguise," in *Proceedings of IEEE Workshop on Applications of Computer Vision*, 2013, pp. 186–192, doi: 10.1109/WACV.2013.6475017.
- [82] G. Goswami, S. Bharadwaj, M. Vatsa, and R. Singh, "On RGB-D face recognition using Kinect," in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2013, pp. 1–6.
- [83] M. Carvalho, B. Le Saux, P. Trouvé-Peloux, A. Almansa, and F. Champagnat, "On regression losses for deep depth estimation," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 2915–2919.
- [84] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014.
- [85] 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*, 2016, pp. 770–778.
- [86] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.
- [87] M. Song, S. Lim, and W. Kim, "Monocular Depth Estimation Using Laplacian Pyramid-Based Depth Residuals," *IEEE Trans. Circuits Syst. Video Technol.*, 2021.
- [88] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3684–3692.
- [89] 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 Prepr. arXiv1907.10326*, 2019.
- [90] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *arXiv Prepr. arXiv1907.01341*, 2019.
- [91] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12179–12188.
- [92] I. Alhashim and P. Wonka, "High Quality Monocular Depth Estimation via Transfer Learning," 2018, [Online]. Available: <http://arxiv.org/abs/1812.11941>.
- [93] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 10691–10700, 2019.
- [94] F. Khan, S. Basak, and P. Corcoran, "Accurate 2D Facial Depth Models Derived from a 3D Synthetic Dataset," in *2021 IEEE International Conference on Consumer Electronics (ICCE)*, 2021, pp. 1–6.
- [95] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library

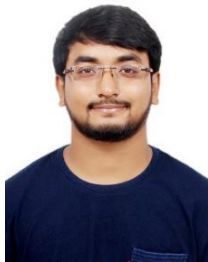
for 3D data processing,” *arXiv Prepr. arXiv1801.09847*, 2018.



FAISAL KHAN earned his bachelor's degree in mathematics from the University of Malakand Chankdara lower Dir in Pakistan in 2015, and his master's degree in mathematics from Hazara University Mansehra in Pakistan in 2017. He is currently working on his PhD at the National University of Ireland in Galway (NUIG). He also works for FotoNation/Xperi. His research focuses on deep neural networks for machine learning applications in computer vision, such as depth estimation and 3D reconstruction.



MUHAMMAD ALI FAROOQ received his BE degree in electronic engineering from IQRA University in 2012 and his MS degree in electrical control engineering from the National University of Sciences and Technology (NUST) in 2017. He is currently pursuing the Ph.D. degree at the National University of Ireland Galway (NUIG). His research interests include machine vision, computer vision, video analytics, and sensor fusion. He has won the prestigious H2020 European Union (EU) scholarship and currently working with NUIG, one of the consortium partners in the Helias (thermal vision augmented awareness) project funded by EU.



WASEEM SHARIFF received his B.E degree in computer science from Nagarjuna College of Engineering and Technology (NCET) in 2019 and his M.S. degree in computer science, specializing in artificial intelligence from National University of Ireland Galway (NUIG) in 2020. He is working as research assistant at National University of Ireland Galway (NUIG). He is associated with Helias (thermal vision augmented awareness) project. He is also allied with FotoNation/Xperi research team. His research interests include machine learning utilizing deep neural networks for computer vision applications, including working with visible, synthetic data, thermal data, and other bio-sensors.



SHUBHAJIT BASAK received his B.Tech. in Electronics and Communication Engineering from West Bengal University of Technology in India in 2011 and his M.Sc. in Computer Science from National University of Ireland Galway in Ireland in 2018. He has more than 6 years of experience as a software developer in the corporate world. He is now pursuing a Ph.D. in Computer Science at Ireland's National University of Ireland, Galway. He also works for FotoNation/Xperi. Deep learning tasks relating to computer vision are among his research interests.



PETER CORCORAN (Fellow, IEEE) is the Personal Chair in Electronic Engineering at the National University of Ireland Galway's College of Science and Engineering. He has been named an IEEE Fellow for his contributions to digital camera technologies, particularly in-camera red-eye correction and facial recognition. He was a co-founder of many start-up firms, including FotoNation, which is now part of the Xperi Corporation's Imaging Division. He has over 600 technical publications and patents under his belt, as well as over 100 peer-reviewed journal articles, 120 international conference papers, and is a co-inventor on over 300 granted US patents. For over 25 years, he has been a member of the IEEE Consumer Electronics Society. He is the Founding Editor and Editor-in-Chief of IEEE Consumer Electronics Magazine.

Appendix D

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

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].

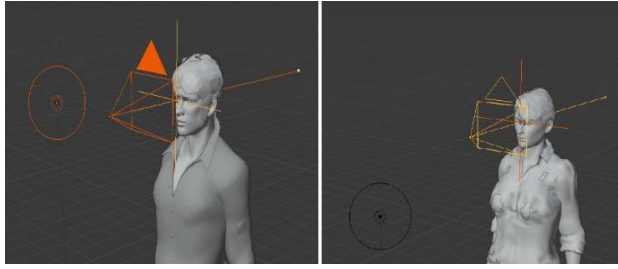


Fig. 4. A sample of the camera and light position with respect to the 3D character.

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].

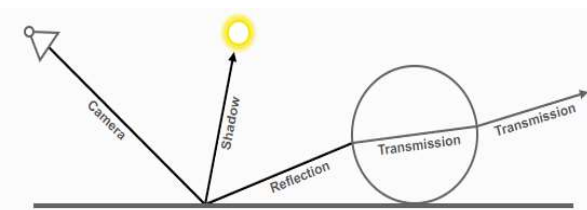


Fig. 7. An overview of the noise control system in Blender.

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.

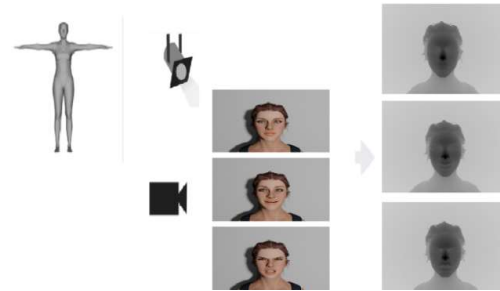


Fig. 8. Rendering configuration in Blender. The left row shows the body shape, light and camera setting; the middle row shows the facial RGB and the last row illustrates the corresponding facial depth image.

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.

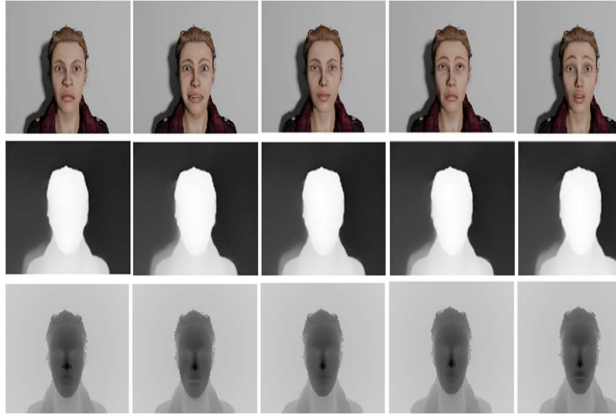


Fig. 11. Sample synthetic RGB images, predicted depth maps by MiDas [20] and corresponding ground truth.

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.

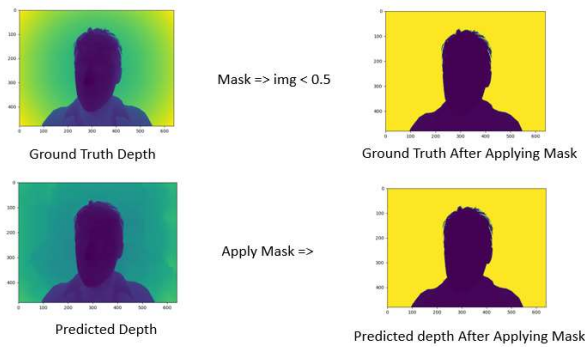


Fig. 12. Ground Truth Depth and Predicted Depth before and applying the mask.

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.

Appendix E

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

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].

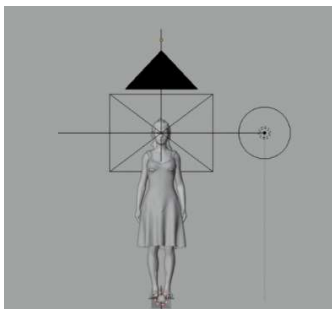


Fig. 4. An example of the 3D character in Blender [8] shows the light location and camera position.

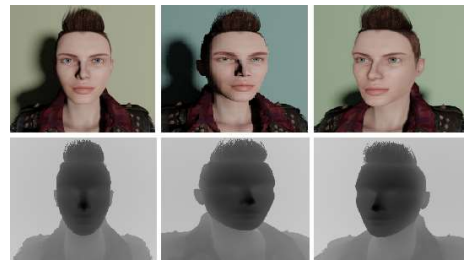


Fig. 5. An example of the head model from various views of the facial model and the corresponding depth information.



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

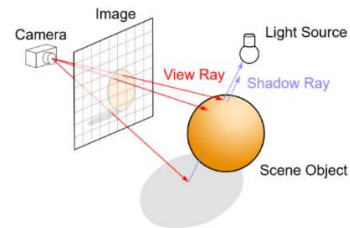


Fig. 7. An overview of the noise reduction method.

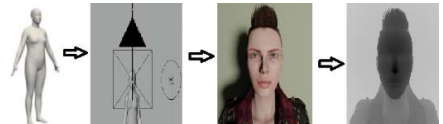


Fig. 8. A simple view of the rendering configuration in Blender [8].



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.

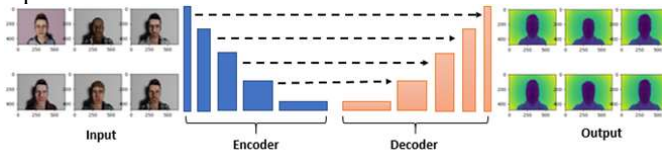


Fig. 13. An example of the proposed network architecture.

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

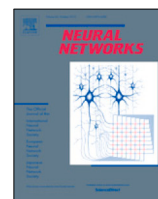
application with the focus on robotics, 3D reconstruction, beautification, autonomous vehicles, natural face modelling and augmented reality.

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*.

Appendix F

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



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

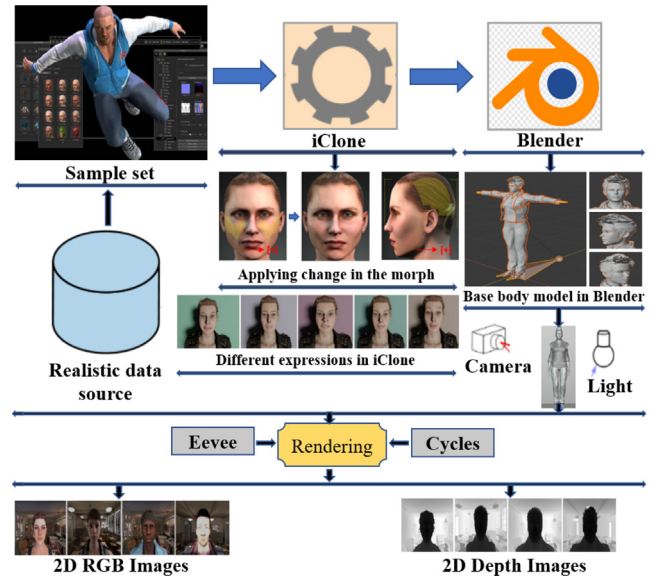


Fig. 1. A schematic representation of generating the synthetic human facial dataset: Samples from the 100 Realistic Head Models, with variation in gender, race, and age. In iClone, changing the morph to create variations to the head models. Importing fully rigged FBX models from iClone to Blender, lighting,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

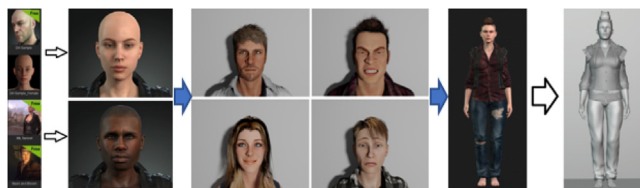


Fig. 2. From left to right: Samples from the 100 Realistic Human Models with variation in gender, race, age and facial expressions followed with a fully rigged FBX model from iClone to Blender with the mesh representation.

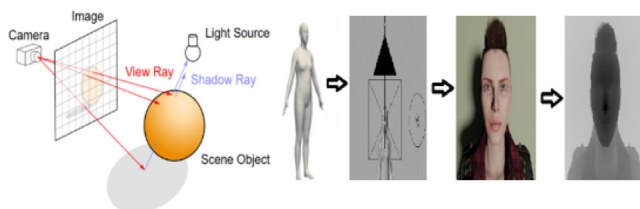


Fig. 3. In Blender, a simplified view of the rendering configuration. The left row shows the body shape, light and camera setting information; the middle row shows the facial RGB image and the last row illustrates the corresponding facial depth image.

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 (Borghi 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.

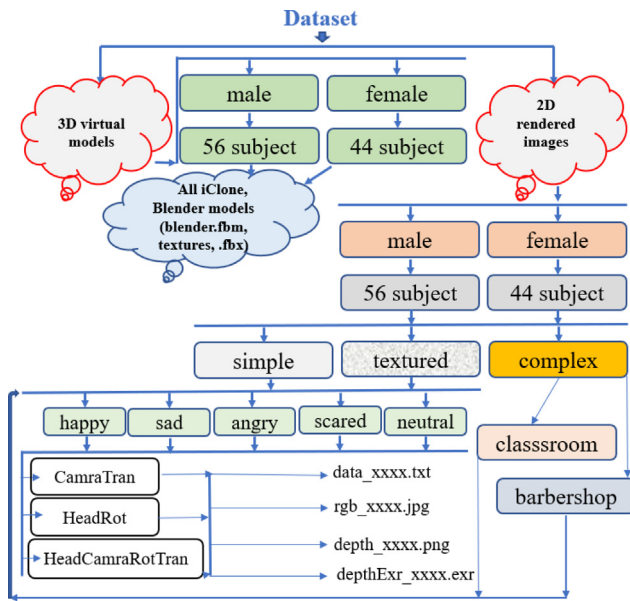


Fig. 5. Dataset organization: The dataset is divided into different folders which correspond to each 'subject' being captured and rendered with RGB images; ground truth depth images.

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

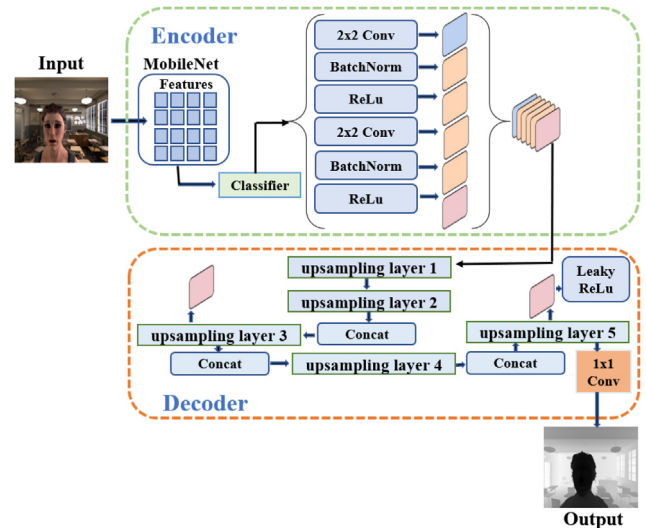


Fig. 6. Schematic diagram of the proposed depth estimation network: A multi-layer Encoder–Decoder network is used to generate accurate facial depth maps based on the MobileNet backbone model.

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

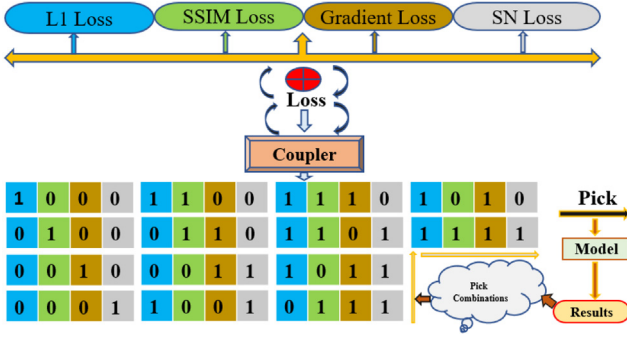


Fig. 7. An illustration of the hybrid loss function composition: A hybrid loss function is introduced through the combination of point-wise loss, gradient loss, surface normal loss, and SSIM loss functions.

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.

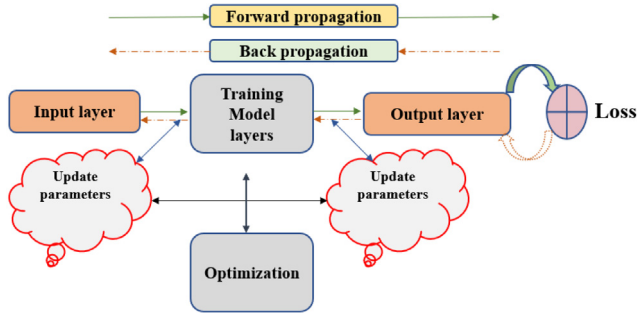


Fig. 8. Overall implementation details of training the proposed model with hybrid loss function.

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:

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

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

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

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

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

$$NRMSE = \frac{RMSE - RMSE_{min}}{RMSE_{max} - RMSE_{min}} \tag{11}$$

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

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

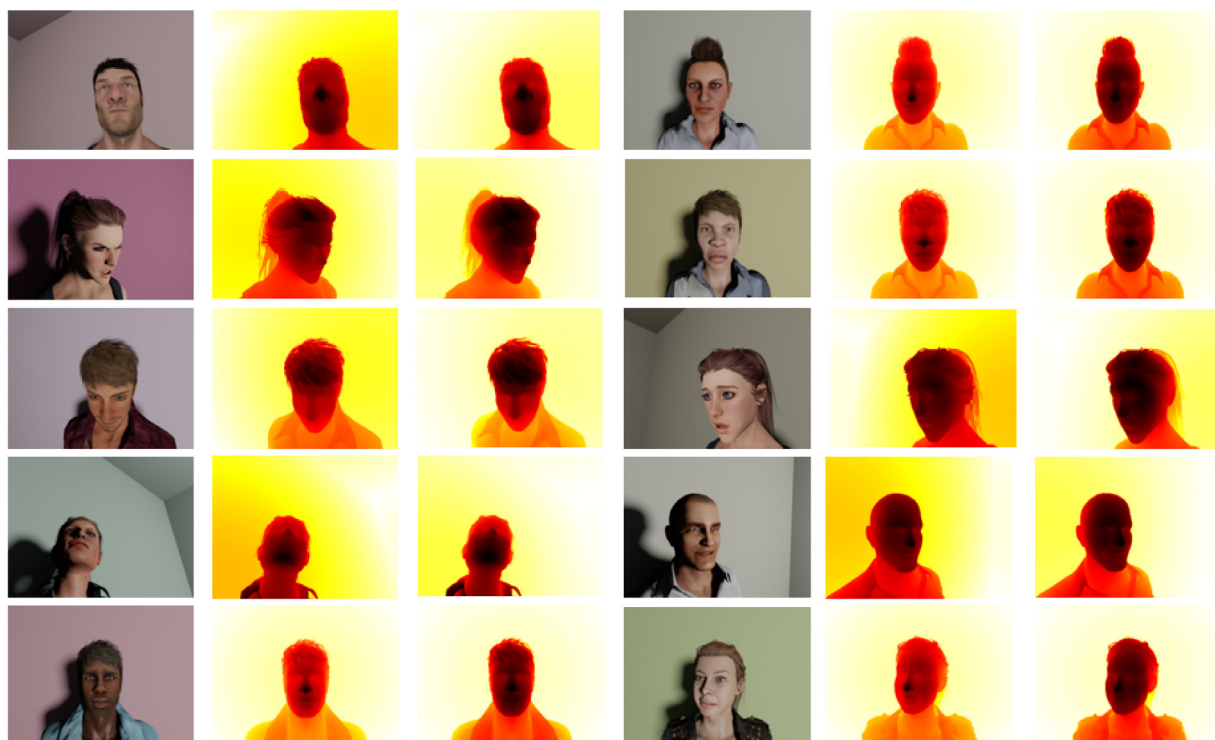


Fig. 9. Qualitative results of the proposed method on a subset of the synthetic human facial dataset that was not used for training or validation. From left to right, input RGB images, ground truth depth images and predicted depth images.

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

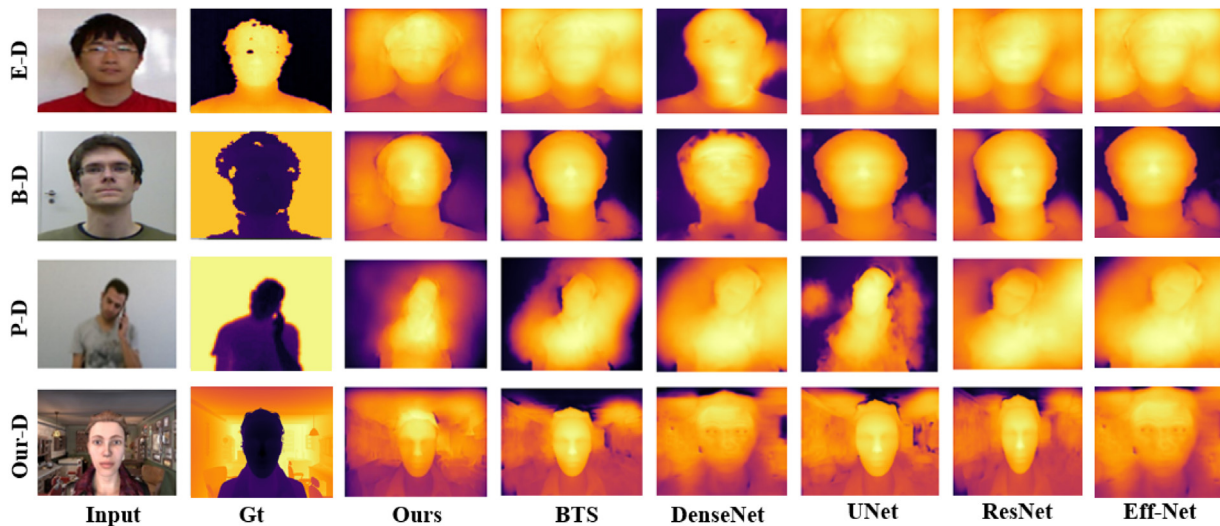


Fig. 10. A qualitative comparison of our approach to the four best competitors: from left to right; (Input: input RGB images; GT: ground truth images; Ours: Our FaceDepth method; BTS (Lee et al., 2019), Ef-Net: EfficientNet-B7 (Alhashim & Wonka, 2018; Wang et al., 2019); Rs-Net: ResNet-50 (Alhashim & 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.

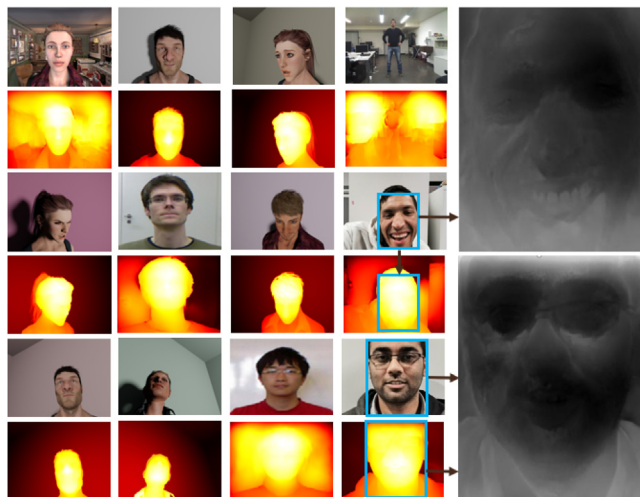


Fig. 11. Results of the baseline model trained using our proposed hybrid loss function and synthetic human facial dataset. The model trained using the hybrid loss function provides more details of local depth structure and higher accuracy at depth boundaries. The test images are a combination of real 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

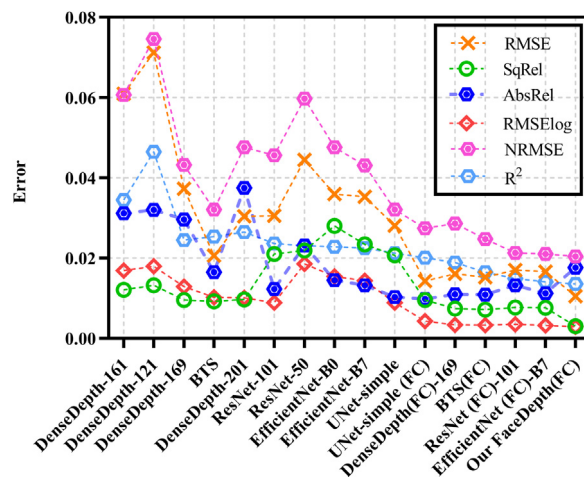


Fig. 12. The relative performance of several technique evaluation metrics (lower is better).

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

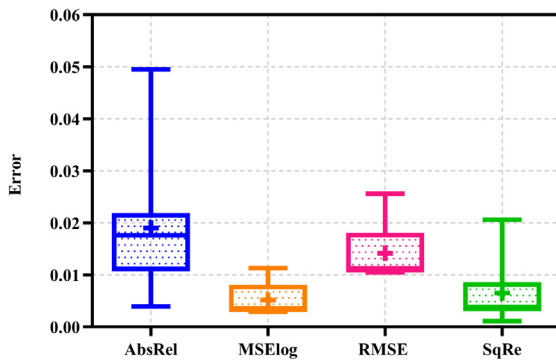


Fig. 13. The FaceDepth method box plot shows the relative performance of various errors.

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., Venturilli, 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.

Appendix G

A Robust Light-Weight Fused-Feature Encoder-Decoder Model for Monocular Facial Depth Estimation from Single Images Trained on Synthetic Data

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Robust Light-Weight Fused-Feature Encoder-Decoder Model for Monocular Facial Depth Estimation from Single Images Trained on Synthetic Data

FAISAL KHAN¹, WASEEM SHARIFF^{1,3}, MUHAMMAD ALI FAROOQ¹, SHUBHAJIT BASAK²
AND PETER CORCORAN.¹, (Fellow, IEEE)

¹Department of Electronic Engineering, College of Science and Engineering, National University of Ireland Galway, Galway, H91 TK33 Ireland

²School of Computer Science, National University of Ireland Galway, Galway H91TK33, Ireland

³Xperi Inc., Galway, Ireland

Corresponding author: Faisal Khan (e-mail: f.khan4@nuigalway.ie).

This work is supported by the College of Science and Engineering, National University of Ireland Galway, Galway, H91TK33, Ireland; and the Xperi Galway Block 5, Parkmore East Business Park, Galway, H91V0TX, Ireland

ABSTRACT Due to the real-time acquisition and reasonable cost of consumer cameras, monocular depth maps have been employed in a variety of visual applications. Regarding ongoing research in depth estimation, they continue to suffer from low accuracy and enormous sensor noise. To improve the prediction of depth maps, this paper proposed a lightweight neural facial depth estimation model based on single image frames. Following a basic encoder-decoder network design, the features are extracted by initializing the encoder with a high-performance pre-trained network and reconstructing high-quality facial depth maps with a simple decoder. The model can employ pixel representations and recover full details in terms of facial features and boundaries by employing a feature fusion module. When tested and evaluated across four public facial depth datasets, the suggested network provides a more reliable state-of-the-art, with significantly less computational complexity and a reduced number of parameters. The training procedure is primarily based on the use of synthetic human facial images, which provide a consistent ground truth depth map, and the employment of an appropriate loss function leads to higher performance. Numerous experiments have been performed to validate and demonstrate the usefulness of the proposed approach. Finally, the model performs better than existing comparative facial depth networks in terms of generalization ability and robustness across different test datasets, setting a new baseline method for facial depth maps.

INDEX TERMS Facial Depth Estimation, Feature Fusion, Encoder-Decoder architecture, Deep learning

I. INTRODUCTION

DEPTH estimation is a crucial challenge that is used in a variety of computer vision applications, including 3D vision [1], 3D face recognition [2], and autonomous vehicles [3] due to the low cost of consumer depth cameras and real-time performances. Raw depth maps, on the other hand, continue to face significant acquisition distortion and detailed corruption. An extensive study has lately been conducted to increase depth accuracy, with the majority of these studies leveraging additional details, such as RGB images or multi-depth maps, for depth map enhancement, while a few employ single depth map enhancement [4]–[8]. Although, few studies focus on facial depth maps [9]. The improvement

of facial depth estimation is an important research topic for rapid and low-cost 3D face applications. When compared to ordinary scenery, human faces contain fine structures. Face recognition and other facial depth applications require features that can be used to distinguish one face from another. This makes it more difficult to refine facial depth. With the advancement of the autonomous industry, it is essential to monitor the driver of a vehicle in order to achieve safety, comfort, and enhanced human-vehicle interactions [10]. As a proof of concept, the depth estimation in the intelligent vehicle's monitoring system is an advanced way to analyze the driver's behaviour in 3 dimensional instead of 2-dimensional environments. Human facial depth maps are one of the most

frequently encountered objects in facial images and are critical for a variety of facial image processing activities. From human facial geometry, the eye separation task in a human facial region is limited to a small range, and thus, using the field of view information from the camera sensors, it is possible to determine the distance between the camera and the subject with reasonable accuracy from a single image frame. A neural network can be trained to estimate depth more accurately by using data that includes face images, towards which this study is aimed. It should be possible for the neural model to understand a considerable measure of the details of human facial structure and properties that can improve the state-of-the-art (SoA) in facial depth map research.

In this paper, the main contribution is to propose a new neural facial depth estimation network that uses a single image and predicts accurate facial depth maps. As compared to the previous facial depth estimation algorithms, this network is significantly smaller in size and less cost-effective, making it ideal for embedded systems and edge-AI applications. Based on the evaluation of four public facial depth datasets, this lightweight network outperforms equal to, or better than SoA across different primary measures. Furthermore, extensive experiments demonstrate the utility and generalization of the proposed network. The rest of the article is organized as follows. Section 2 discusses related research that has been conducted in relation to the proposed method. Section 3 presents and discusses the proposed neural facial depth estimation network for generating facial depth maps. A large and diverse synthetic dataset is used in the training phase, and a series of experimental comparisons, evaluations of the presented approach against the existing SoA approaches and results for facial depth maps are discussed in sections 4 and 5. In Section 6, the results of this research work are briefly discussed. Section 7 addresses the challenges, future trends, and improvements, while Section 8 summarizes the research.

II. RELATED WORKS

Interpreting spatial relationships within a scene involves estimating depth maps. As a result, such relationships assist in the creation of stronger representations of objects and their surroundings, which can lead to advancements in existing recognition tasks as well as the development of new applications like 3D modelling. With only a single RGB image as input, the purpose of monocular depth estimation is to estimate the depth value of each image pixel or derive a depth map. There has been a lot of effort put into the past to estimate depth using stereo images, as well as progress being made by researchers in monocular depth estimation due to the advancement in convolutional neural networks [7], [11]–[15]. However, monocular facial depth estimation research has recently gotten attention [9]. Monocular depth estimation employs a single camera to acquire an image or video sequence and requires no more complex equipment or professional techniques. It has a broad range of application requirements due to the availability of only one camera in the

majority of application scenarios. As a result, the need for monocular depth estimation has increased in recent years, [15]. Facial depth estimation has many applications and approaches using both conventional and traditional methodologies [8]. Using the feature extraction methods, There are many SoA potential solutions to predict facial depth [16]–[22]. Facial feature extraction depth maps can help in the advancement of facial depth tasks. On the other hand, with feature fusion methods, rich internal information of the depth, and compressed reconstructions of integrated features can be generated after dimensionality reduction. There are several approaches that are offered in different tasks: [23]–[25].

In recent years, facial depth estimation methods have been proposed for various tasks. Authors in [26], devised a face recognition system in which Fully Convolutional Network (FCN) seeks to recover depth from an RGB image while Convolutional Neural Network (CNN) preserves individual subject separability. In [21] proposed a face depth estimator with conditional generative adversarial networks (GAN). They created a GAN-based approach for estimating depth maps from single-face images. This method also concluded that the conditional Wasserstein GAN structure is the most reliable technique using GAN-based networks. Authors in [27] used an unsupervised approach to estimate depth with 3D face rotation and replacement by implying the depth of an input image's facial key points. In [28] proposed a GAN-based technique to produce robust facial depth estimation. Further [9] proposed a GAN-based technique via segmentation and mask-guided attention network for face depth estimation. Recent research has also revealed that, in addition to colour and deformation, the depth of Ground Truth (GT) of a face can be used to discriminate between real and synthetic faces. It is a strategy worth researching to increase the label information by utilizing estimated depth image labels instead of coding labels. The authors in [30], suggested an auxiliary supervised technique that uses estimated face depth information to expand label information.

In addition, there has been significant research towards generating 3D synthetic facial depth estimation methodologies. In [31], authors provided realistic 2D facial depth models obtained from a 3D synthetic dataset. The authors also suggested a benchmark dataset, as well as a CNN-based architecture for predicting depth from a 2D image in [32]. The authors in [8] offered a comprehensive review of monocular facial depth estimation, including types of approaches that have been and can be used in past, current, and future research.

III. LIGHTWEIGHT ENCODER-DECODER BASED FACIAL DEPTH ESTIMATION MODEL

Numerous consumer applications, such as robots, augmented reality, and automated driver monitoring systems, can benefit from neural facial depth estimation networks constructed from single image frames. Conventional approaches utilize fully connected layers, which complicates the models and

necessitates additional memory, making them unsuitable for deployment on consumer devices and they suffer from issues such as information loss that leads to holes in depth-images. On the other hand, many Deep Learning (DL) techniques have recently been presented, and they have shown considerable progress in solving the fundamental ill-posed problem of depth estimation. This article describes the procedure for constructing depth maps from a single-frame face image that makes use of the input RGB face image and the corresponding GT depth utilizing neural networks.

Keeping a simple model architecture in mind that can be used for consumer devices for real-life applications, the model applied in this research work automates the collection of optimal parameters and a less number of parameters size thus reducing model complexity during the training procedure. The proposed model is more computationally efficient than the current SoA facial depth maps models and shows performance equal to, or better than SoA when tested across 4 public depth datasets. The performance of the proposed CNN model is evaluated with SoA networks, and different encoders including EfficientNetB0, ResNet-101, and DenseNet-169 are compared.

A. NETWORK ARCHITECTURE

This section describes the proposed neural facial depth network for the mechanism of single-image facial depth maps, as well as the suggested loss function for optimizing the procedure over the training data. The framework's general architecture is demonstrated in Fig. 1. To obtain high-quality facial depth maps, researchers usually create deeper networks with additional parameters and constraints, which need additional computation complexity and hence do not match the real-time requirements of real-time applications. As a result, the authors sought to develop a lightweight neural facial depth model capable of real-time facial depth prediction while maintaining prediction accuracy equal to or better than current SoA networks.

1) Encoder Model

The proposed decoder for reconstructing facial depth residuals is coupled to the network's pre-trained encoder ResNet18 [33] and the main feature of the network have been described in Fig. 1 and Fig. 2. In the encoder process, the model consists of 22 layers including eight parts: convolutional layers 1-5, a global average pooling (AP) layer, and a fully connected (FC) layer. The initial features are corrected in the channel dimension to increase the model's intensity of learning features, enabling the model to automatically pick up on the key characteristics of various channels. The global average pooling layer is then used in place of the fully-connected layers to decrease model parameters, speed up model convergence and enhance the accuracy of the model.

2) Decoder Model

The presented model's encoder takes the input image to block features of various sizes. A lightweight and efficient decoder

is utilized to recover the bottleneck features in order to extract the estimated facial depth map [6]. The better performance is demonstrated experimentally to be due to the training process. Additionally, the model achieves higher performance by utilizing much fewer convolutional and bilinear upsampling layers in the decoder. To begin, convolution is employed to lower the channel dimension of the bottleneck feature, hence avoiding the complexity of the algorithm. Following that, a series of bilinear upsampling layers are utilized to enhance the size of the features. Lastly, two convolution layers and a sigmoid function are used to the output to estimate the facial depth map. Additionally, the depth map is scaled by the value of the maximum depth to give the depth in meters. A skip connection is introduced to the proposed fusion module in order to make better use of the precise details of the local structures.

B. LOSS FUNCTION

The objective of the facial depth estimation problem is to design a function that accurately predicts the depth of an input image. (L_{silog}) seems to be the most frequently used and the best choice loss function in the training process because it is more useful for reducing errors in facial depth estimation. The network's learnable parameters are optimized focusing on the loss function, which implements correctly scaling the loss function's range to enhance convergence and training output results while attempting to put a stronger focus on $lamda$ -based error variance reduction, resulting in a Silog loss function. [35](Lee, Han, Ko and Suh, 2019b) (L_{silog}) is defined:

$$L_{si}(y_i, y_i^*) = \frac{1}{n} \sum_i (\log(y_i) - \log(y_i^*))^2 - \frac{\hat{\lambda}}{n^2} \left(\sum_i \log(y_i) - \log(y_i^*) \right)^2 \quad (1)$$

where $\hat{\lambda}$ is the balancing factor, and n is the pixel count. Through a rewrite of the equation. 1:

$$L_{silog}(y_i, y_i^*) = \frac{1}{n} \sum_i (\log(y_i) - \log(y_i^*))^2 - \left(\frac{\hat{\lambda}}{n} \sum_i \log(y_i) - \log(y_i^*) \right)^2 + (1 - \hat{\lambda}) \left(\frac{\hat{1}}{n} \sum_i \log(y_i) - \log(y_i^*) \right)^2 \quad (2)$$

It's a sum of the variance and a balanced square average of the error in log space. As a result, founding a larger $\hat{\lambda}$ imposes a greater focus on limiting error variance, Also, it is found that adjusting the loss function's range properly increases convergence and the overall training result. In log space, the combined Silog loss is defined as:

$$L_{silog}(y_i, y_i^*) = \alpha \sqrt{L_{silog}(y_i, y_i^*)} \quad (3)$$

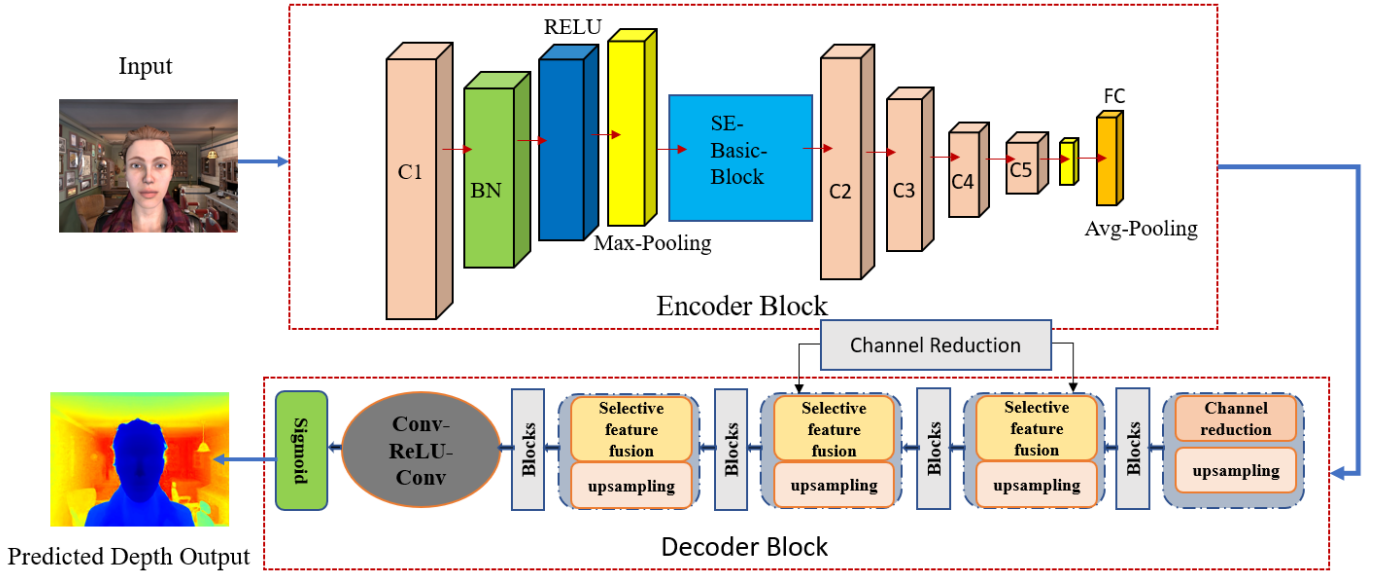


FIGURE 1. The proposed approach for monocular facial depth estimation’s architectural shape. The encoder has the Resnet18 network, and the proposed decoder architecture’s primary components along with channel reductions, skip connections and feature fusion modules.

Layers	Output Size	Layer parameters
C1	$112 \times 112 \times 64$	$7 \times 7, 64, \text{stride } 2$
		$3 \times 3 \text{ maxpool, stride } 2$
C2	$56 \times 56 \times 64$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
C3	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
C4	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
C5	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
AP	$1 \times 1 \times 512$	$7 \times 7 \text{ AP}$
FC	2	$512 \times 2 \text{ FC}$
softmax	2	FC

FIGURE 2. The Encoder model’s detailed structure is used in the proposed method.

IV. EXPERIMENTS

The experimental results are discussed and summarized to demonstrate the effectiveness of the proposed approach in comparison to SoA methods. The proposed model is trained on a synthetic facial depth dataset and then compared to four real datasets. Numerous comparisons have been conducted, as well as evaluations of its accuracy and computational footprint.

The studies show that a network trained on a wide and diverse set of images, along with a decent training technique, produces SoA performance in many situations, particularly for faces. The zero-shot cross-dataset transfer technique is used to show the method’s effectiveness.

A. IMPLEMENTATION DETAILS

The model for estimating the facial depth is trained with the PyTorch DL framework. For training and testing, the data was divided into 0.8 and 0.2 ratios, and the model was evaluated against four publicly available datasets. We employ the one-cycle learning rate technique with an Adam optimizer in all of the experiments. The learning rate increased by 0.9 during the first half of the total iterations from $3e-5$ to $1e-4$ following a poly LR scheduling and then falls by a factor of 0.9 from $1e-4$ to $3e-5$ in the second half. On a workstation equipped with NVIDIA 2080ti GPUs, the total number of epochs is set to 50 with a batch size of 16. There are around 12.06 million trainable parameters in the proposed model.

The Root Mean Square Error (RMSE), the log Root Mean Square Error (RMSE (log)), the Absolute Relative difference (AbsRel), the Square Relative error (SqRel), and the Accuracies are used to perform the evaluations (Equation (4-10)). With a 50% probability, the following procedures are utilized for data augmentation: horizontal flips, random brightness(0.2), contrast(0.2), gamma(20), hue(20), saturation(30), and value(20). We use $p = 0.75$ for vertical CutDepth with a probability of 25%.

Fig. 3 depicts the whole experimental implementation details including training, evaluation, and testing of the proposed model using a synthetic facial depth dataset. First, the model is trained with a synthetic facial depth dataset and then evaluated and tested with four real depth datasets (mentioned in section 4.3) against SoA DL methods. The proposed model uses a single frame RGB image and corresponding GT depth image as training data for the convolution layer to extract features. CNN uses a weight-sharing method that significantly reduces the number of parameters, greatly enhancing the model’s performance.

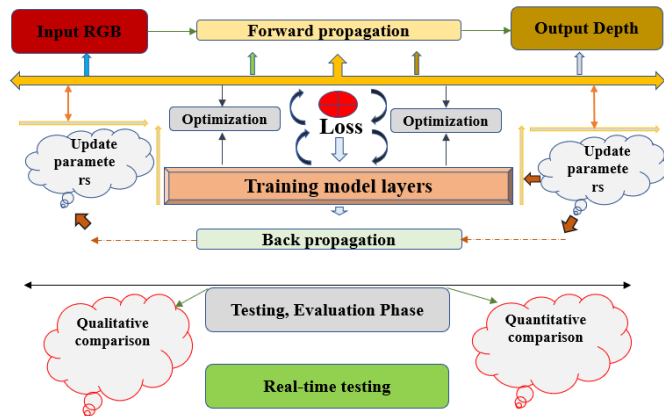


FIGURE 3. The implementation details of the suggested neural network training, evaluation and testing procedure.

B. TRAINING DATASET

The proposed LEDDEPTH model is trained on the synthetic human facial depth dataset and evaluated with four other test datasets for rigorous comparison with other SoA models which includes BTS [35], Densdepth [34], UNet-simple [36], ResNet-101 [37], EfficientNet-B0 [38], MiDaS [39]. Further details of the training dataset are presented in subsection 4.2.1.

1) Synthetic Human Facial Depth dataset [37]

There are a considerable number of high-quality 3D face models in the Synthetic Human facial depth datasets as well as 2D RGB and pixel-accurate ground truth depth images. Character Creator is accustomed to using 100 real-world head models to create a series of virtual human avatars. The models' textures and topologies are adjusted to increase the number of possible samples. After loading the models into iClone, 5 distinct facial expressions are incorporated into the data. Importing the FBX files of the iClone models and their associated mesh, textures, and animation keyframes into Blender is the final step in the process.

All Blender models have been rotated in order to get the proper head position. Thereafter, the FBX models are imported into Blender and adjusted to the reference frame. Lights and cameras are used in the environment to mimic the real-world environment, and their attributes are then altered accordingly. The camera lens's near and far clips have been set to a distance of 0.01 meters and a maximum of 5 meters. 60 degrees of FOV is achieved by adjusting both the sensor's resolution and the sensor's field of view (FOV). The final effect is attained by configuring the render layer's RGB and Z-pass outputs in the compositor. In posture mode, the joints of the head and shoulders are detected, the head mesh pivots these bones, and frames are saved to carry out the rotational movement.

Finally, all frames are rendered in order to produce the RGB and depth images needed for final rendering. With the help of the Python code available in the Blender application, the head position (yaw, pitch, and roll) has been created. A

640x480 pixel RGB image is created and saved in jpg format for each frame. while the depth data is saved in (.exr format). A text file (.txt) containing each frame's head positions is also stored. The Cycle Rendering Engine, integrated with Blender, renders each 2D image in about 26.3 seconds on average. Using Cycle Rendering Engines is used to track the progress of the rendered scenes. There are around 3,500k frames in the entire collection, and each model receives about 3.5k 2D images. The following link has detailed information about the dataset. (<https://dx.doi.org/10.21227/ath9-br59>) booktabs

C. TEST DATASETS

There are numerous datasets available for estimating facial depth, each with a unique type and depth range. Four datasets are chosen for the diversity and quality of their source data for facial depth map predictions. Those include the following: Pandora [40], Eurecom Kinect Face [41], Biwi Kinect Head Pose [41] and Synthetic Human Facial Depth [37] test dataset for testing and evaluations purposes.

1) Pandora

The Pandora dataset is utilized for a variety of purposes, including estimating head pose, head centre localisation, depth estimation, and shoulders pose estimation. It includes 250K full-resolution RGB images and their corresponding depth images.

2) Eurecom Kinect Face

The dataset contains multi-model facial images of 52 individuals, 38 of who are male and 14 of whom are female, collected with the Kinect sensor. It includes nine distinct states of facial expression, occlusion, and illumination, including grin, eye obstruction, mouth, light and sheet, moderate, open mouth, and left-right profiling.

3) Biwi Kinect Head Pose

Contains 15k images of 20 subjects taken with the Kinect sensor as the subjects' heads were freely moved around across each side. Each frame contains RGB and depth images, as well as the head's 3D position and rotation angles.

D. EVALUATION METRICS

To interpret the data, a widely known assessment procedure with several evaluation indicators is being used: The root mean square error (RMSE), the log root mean square error (RMSE (log)), the absolute relative difference (AbsRel), the square relative error (SqRel), the accuracies, the normalized root mean square error (NRMSE), and the R-squared. All of those are as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n \|y_i - y_i^*\|^2} \quad (4)$$

$$RMSE_{Log} = \frac{1}{n} \sum_{i=1}^n \|\log(y_i) - \log(y_i^*)\|^2 \quad (5)$$

TABLE 1. Comparison of various depth maps methods with the proposed method LEDDEPTH, BTS [35], Densedepth [34], UNet-simple [36], ResNet-101 [37], EfficientNet-B0 [38], MiDaS [39], DPT [15], LapDepth-Face [8], FaceDepth [37] on synthetic human facial depth dataset [37]

No.	Methods	AbsRel	SqRel	RMSE	RMSElog	$\delta_1 < 1.25$	$\delta_2 < 1.25^2$	$\delta_3 < 1.25^3$
1.	DenseDepth-161	0.0296	0.0096	0.0373	0.0129	0.9890	0.9920	0.9981
2.	ResNet-101	0.0123	0.0210	0.0306	0.0089	0.9938	0.9960	0.9980
3.	BTS	0.0165	0.0092	0.0206	0.0102	0.9830	0.9943	0.9956
4.	EfficientNet-B0	0.0145	0.0280	0.0360	0.0154	0.9912	0.9934	0.9978
5.	UNet-simple	0.0103	0.0207	0.0281	0.0089	0.9960	0.9956	0.9987
6.	MiDaS	0.0146	0.0204	0.0356	0.0323	0.9665	0.9902	0.9983
7.	DPT	0.0156	0.0106	0.0394	0.0184	0.9567	0.9646	0.9943
8.	LapDepth-Face	0.0145	0.0041	0.0204	0.3614	0.9545	0.9857	0.9958
9.	FaceDepth	0.0176	0.0030	0.0205	0.1252	0.9642	0.9849	0.9951
10.	LEDDEPTH	0.0113	0.0025	0.0203	0.1172	0.9888	0.9961	0.9967

$$AbsRel = \frac{1}{n} \sum_{i=1}^n \frac{\|y_i - y_i^*\|}{y_i^*} \quad (6)$$

$$SqRel = \frac{1}{n} \sum_{i=1}^n \frac{\|y_i - y_i^*\|^2}{y_i^{*2}} \quad (7)$$

$$Accuracies = \% \text{ of } y_i \max\left(\frac{y_i}{y_i^*}, \frac{y_i^*}{y_i}\right) = \delta < thr \quad (8)$$

$$NRMSE = \frac{RMSE - RMSE_{min}}{RMSE_{max} - RMSE_{min}} \quad (9)$$

$$R^2 = 1 - \frac{\sum_{m=1}^n (y_i - \bar{y}_i^*)^2}{\sum_{i=1}^n (y_i - \bar{y}_i^*)^2} \quad (10)$$

Where y_i^* is the GT, \bar{y}_i^* is the mean of the GT and y_i is the predicted depth of the pixel i , n represents the overall number of pixels, while thr denotes the accuracy threshold.

V. RESULTS AND COMPARISONS TO PRIOR WORK

The results of the proposed approach are shown in Fig. 4 and Table 1. The performance of the proposed facial depth estimation model is evaluated with the SoA methods BTS [35]; MiDaS [39]; DPT [15]; LapDepth-Face [8] on the synthetic human facial dataset [37]. The network achieves SoA performances in the evaluation metrics SqRel, RMSE and δ_2 . For depth map estimation RMSE is considered the most focal metric for loss estimation thus measuring the performance evaluation of the depth architectures. As can be observed from Table 1, the proposed architecture outperforms other SoA Depth models having the lowest RMSE value.

In the evaluated matrices in Table 1, it can also be observed that the Unet-simple model performs better or is comparable to the suggested model in AbsRel, RMSElog, and δ_1 . The main reason for these results is that the model was trained across the entire image first before being applied to the Facial crop (FC) for evaluating errors in the face region. In other words, the depth has been masked within a 50-centimetre range from the camera so that the results can only be evaluated on the facial region of the images.

The results, as shown in Fig. 5, display high-level detail and constancy, showing that the suggested method performs better at estimating facial depth maps. Note: due to the fact

that the MiDaS network was built to predict inverse depth, the predicted images differ from those of other SoA. Fig. 7 demonstrate the proposed model’s qualitative results on real data and synthetic data compared to SoA techniques. The model outperforms the cutting-edge techniques and sets a new SoA for facial depth estimation. According to the comparison study Table 1 and Fig. 5, the proposed LedDepth method performed best in terms of accuracy and depth range when compared to other SoA approaches. On a synthetic human facial dataset, the network achieved 0.0203 RMSE and 0.9986 threshold accuracy. To the SoA approaches, the suggested lightweight network structure has less parameters and complexity and can be seen from Table 2 and Fig. 6, which provides a full comparative analysis in terms of the number of parameters and computational complexity.

A. QUALITATIVE RESULT

In this subsection, the authors compare qualitative results from the proposed model to SoA approaches. A comprehensive analysis of the proposed method to the four best-performing methods is shown in Fig. 5 and Fig. 7. The suggested model results show better information and consistency, as shown in Fig. 7, proving that the network works better at facial depth estimation.

The model outperformed SoA quantitatively and qualitatively in testing using four datasets and formed a new SoA for facial depth maps. Table 1, Table 2, and Fig. 7 illustrate some of the results.

According to the analyses, the presented scheme significantly outperforms other SoA methods on the basis of consistency and depth range. On the synthetic human facial data, the neural framework obtained a SqRel of 0.0025, RMSE of 0.0203 and a threshold accuracy of 0.9961 as can be seen in Table 1 (row 10).

Additionally, as demonstrated in Table 2 (row 10), the suggested method has a much smaller memory footprint and higher computational efficiency when compared to previous SoA methods. At 25.32 G-MACs per image, this technique enables real-time prediction of single image face depth. Although the LedDepth model has fewer parameters than other SoA, the design principle and simple encoder-decoder stages make it computationally less expensive and can be used for

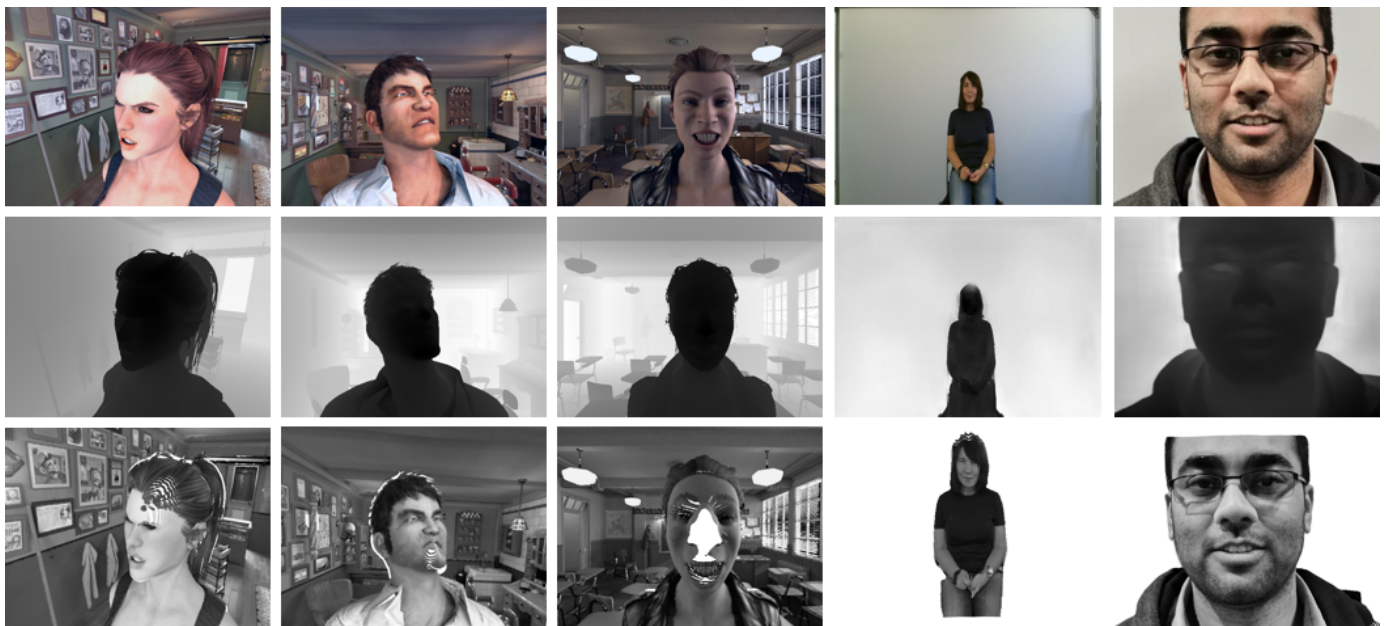


FIGURE 4. The suggested method was evaluated qualitatively using a sample of the synthetic human facial data that was not utilized for training or validation. The first row consists of input RGB images, the second row consists of corresponding predicted depth images, and lastly their rendered point clouds from a novel viewpoint. Point clouds rendered via Open3D [43]

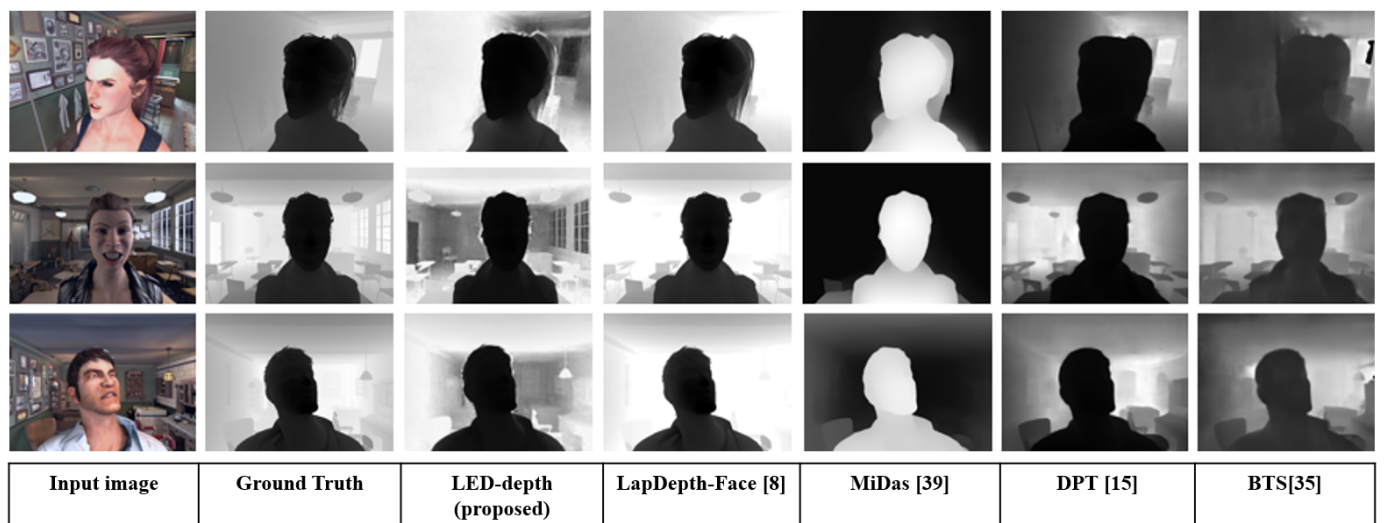


FIGURE 5. Qualitative results of facial monocular depth estimation algorithms on the synthetic human facial dataset.

consumer devices.

The following Table 2 summarizes the characteristics of the models for predicting facial depth maps of a single image frame that have been studied and compared (ED: Encoder-Decoder; F: Trained on the synthetic human facial dataset). According to the test results, DPT [15]; MiDaS [39]; LapDepth-Face [8]; BTS [35] and FaceDepth [37] techniques can build high-resolution facial depth maps with comparable accuracy but are computationally expensive and require a large amount of memory. On the other hand, LedDepth significantly reduced computation time and memory footprints, making it suitable for both high-quality and low-cost single-

image facial depth estimation (Table 2 and Fig. 7).

VI. DISCUSSION

This research proposes a neural model for facial depth estimation and compares its performance to that of current SoA algorithms. Compared to other SoA techniques, the framework proposed has a significantly smaller network size, a smaller number of parameters, and equal or better computing complexity (only less than UNet-simple [36]). It can be noted that when compared to the proposed LEDDEPTH approach, the FaceDepth method in Table 2 is superior in terms of computational complexity, however, the qualitative results

TABLE 2. Properties of the studied methods with the proposed method LEDDEPTH, (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 (GMac)
BTS [35]	640×480F	ED	Adam	46.60M	640×480F	0.0001/50	69.23
DenseDepth-169 [34]	640×480F	ED	Adam	42.60M	320×240F	0.0001/20	66.12
ResNet-101 [37]	640×480F	ED	Adam	68.00M	640×480F	0.0001/25	101.27
EfficientNet-B0 [38]	640×480F	ED	Adam	80.40M	640×480F	0.00001/20	113.44
UNet-simple [36]	640×480F	UNet	Adam	17.27M	640×480F	0.001/20	188.04
FaceDepth [37]	640×480F	ED	Adam	14.42M	320×240F	0.0001/50	16.41
MiDaS [39]	384×384F	CNN	Adam	105.00M	384×384F	0.0001/60	104.00
DPT [15]	384×384F	Transformer	Adam	112.00M	384×384F	0.00001/60	107.00
LapDepth-Face [8]	512×416F	ED	Adam	73.00M	512×416F	0.00001/50	90.85
LEDDEPTH (Proposed)	640×480F	ED	Adam	12.06M	640×480F	0.0001/50	25.32

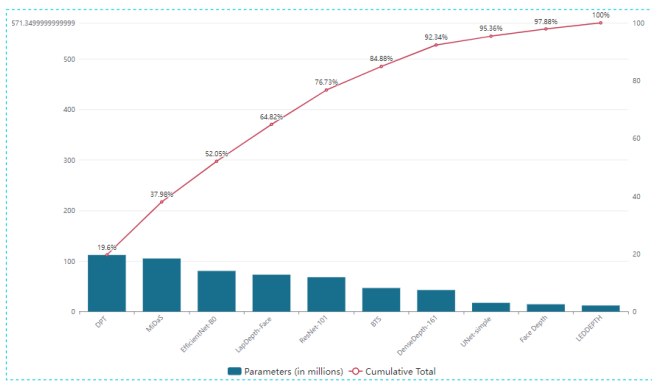


FIGURE 6. The comparison of parameters and their cumulative sum. The proposed LEDDEPTH model contains much less parameters, as shown by the cumulative percentage.

and evaluation metrics are superior to the FaceDepth method in Table 1. In comparison to the existing FaceDepth, the suggested LEDDEPT performed best in terms of accuracy and depth range and can improve its performance in different testing scenes.

The usefulness of the performance is related to the neural model training strategy, which chooses an appropriately optimal loss function by utilizing a synthetic human facial dataset with pixel-accurate ground truth depth information.

As seen in Table 1 and Table 2, the suggested model performs well on the majority of evaluation criteria (Table 1 row 10), which the authors explain to the proposed design and improved depth-specific training methods. Additionally, the suggested neural model outperforms recently published SoA algorithms with fewer parameters (Table 2 bluerow 10). This indicates that the integration of the encoder and the suggested simple decoder clearly contributes significantly to the fast obtaining of accurate facial depth maps. Fig. 7 depicts the visible results. As illustrated in the figure, the model accurately estimates facial depth values for the sample images and is more robust to changing illumination circumstances than other methods. In terms of generalization, DPT and MiDaS performed well in some test images for long-range recognition, however, the proposed LEDDEPTH technique performed better for short-range attribution, particularly for

facial regions.

We perform all tests and evaluations of the SoA on a set of datasets that were never seen during training for both real imaged and synthetic datasets and the results are evaluated using Equations (4-10).

Fig. 8 shows a visual representation of the three different loss function comparisons of the proposed model with two SoA depth networks (BTS and LapDepthFace) It is obvious that the suggested method achieves good performance by minimizing errors over a large number of test datasets when compared to other SoA algorithms. By selecting an appropriate loss function and a pixel-accurate synthetic facial depth dataset, the algorithm is able to decrease error while having a small number of parameters and equal or less computation complexity.

Furthermore, The proposed model is converted to ONNX and it can be used for deployment in embedded systems and in Edge-AI applications. ONNX is a freely available format for encoding deep neural networks. With ONNX, Application developers can more quickly integrate models between SoA packages and determine the ideal mix for their needs. A community of contributors contributes to the development and support of ONNX. Lastly, the release of the code utilized in this study and the publicly available training dataset, as well as the corresponding ONNX transformations, will aid future research in fields like as 3D facial reconstruction, perception, and characterization. https://github.com/khan9048/Facial_Depth_Maps_from_Single_Images

VII. CHALLENGES AND TRENDS

Monocular depth estimation based on DL has been widely researched and advanced during the previous decades. Nevertheless, much more work is required to overcome the limitations, particularly in the area of facial depth estimation.

To enhance the accuracy of depth maps, the majority of studies have concentrated on the layers of neural models, which increases the capacity of the space model and memory consumption. In multi-task neural depth methods for monocular facial depth maps usually use numerous sub-networks to execute distinct sub-tasks, which also increases computations and memory requirements. Typically, most of the monocular

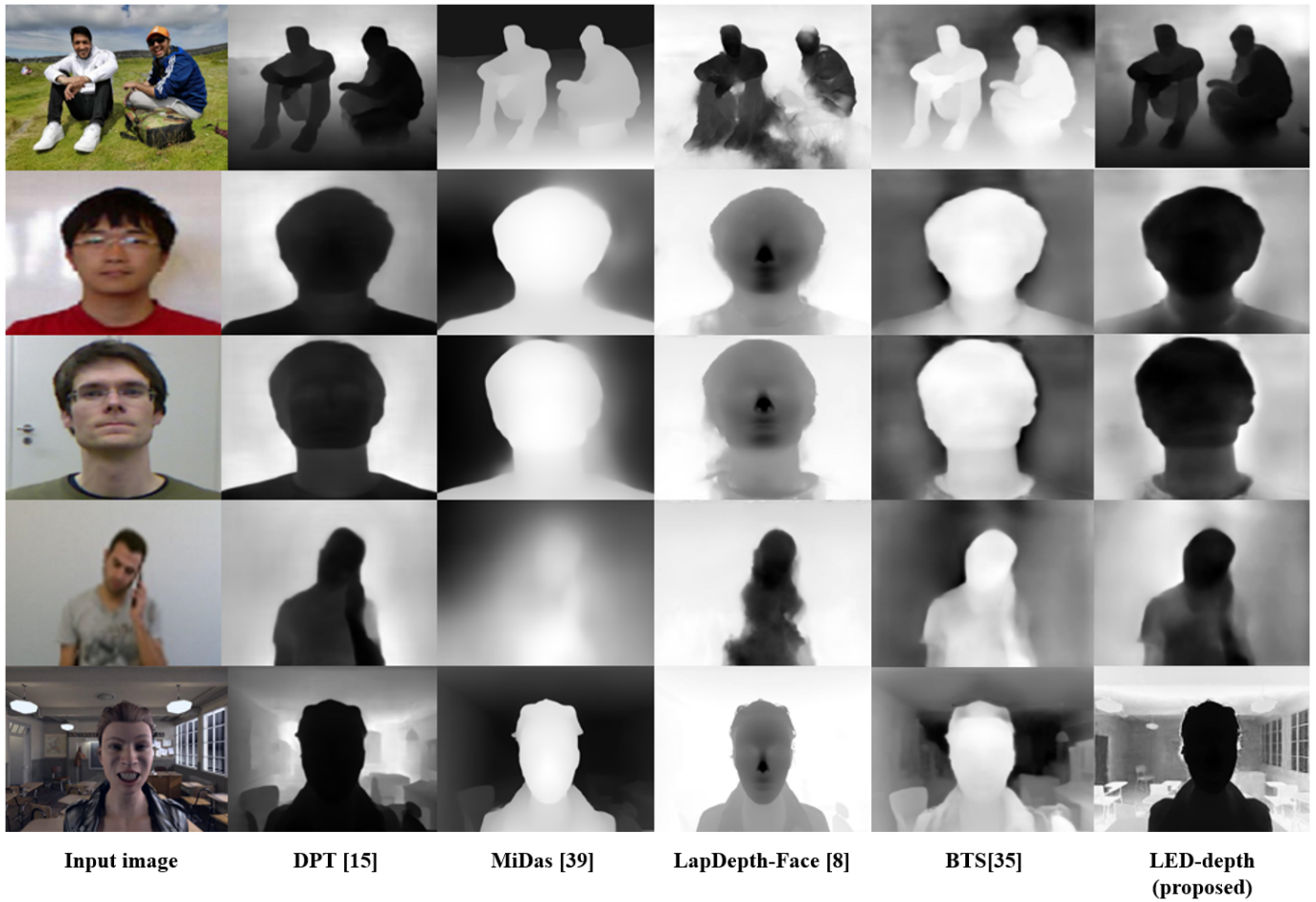


FIGURE 7. A qualitative analysis of our technique in relation to the four SoA methods applied to different datasets (From below:- Synthetic human facial dataset [37]; Pandora dataset [40]; Eurecom Kinect Face dataset [41]; Biwi Kinect Head Pose dataset [42] and an image taken from iPhone 13 pro.

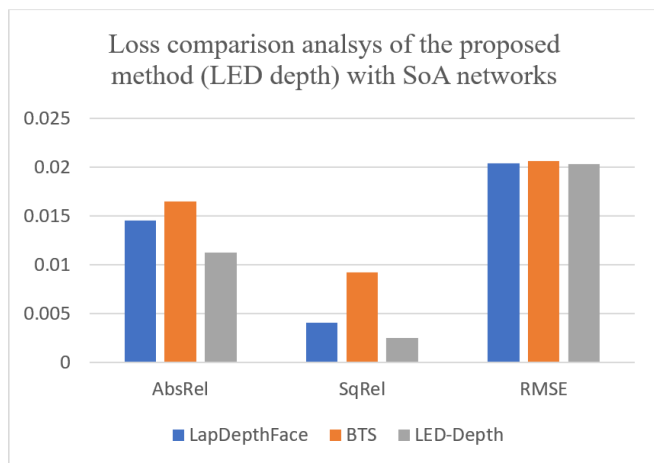


FIGURE 8. The three different evaluations errors metrics: AbsRel, SqRel and RMSE comparison between the proposed network with two SoA networks.

facial depth estimation networks are encoder-decoders with complex structures. After numerous levels of information computation, the depth characteristics are significantly degraded, leading to decreased estimated depth maps that do

not fulfil the practical requirements of the application.

This section covers the major issues and discusses potential directions for monocular facial depth estimation research that can help the researchers in further developments.

A. HIGH-RESOLUTION DEPTH MAP OUTPUT

Facial depth estimation is a critical phase in the evolution of real-world applications such as augmented reality (AR) and virtual reality (VR), and it imposes a great deal of importance on the depth maps accuracies. Nonetheless, the quality of the anticipated facial depth is often limited in most contemporary algorithms in order to maximize computational effectiveness. At the current, research studies are enhancing the super-resolution of depth images using colour image super-resolution frameworks. However, how to properly produce a high-resolution facial depth map remains an open question.

B. REAL-TIME PERFORMANCE

The fundamental module of SLAM is image depth maps, which are tightly coupled with industrial applications such as autonomous driving. As a result, practical applications required pixel-accurate depth map performance. However,

in order to achieve high-quality depth maps, researchers frequently design deeper networks with more parameters and requirements, which requires more computation time and thus does not meet the real-time requirements of real-time applications. Thus, a future research area will be to determine how to use lightweight neural depth models for real-time depth prediction while maintaining prediction accuracy.

C. INTEGRATION AND OPTIMIZATION OF THE NETWORK FRAMEWORK

While it is possible to combine or build a network that can learn both facial depth and segmentation in DL facial depth estimation research, this remains a distinct research field. To learn several tasks, such as face depth maps or segmentation or depth features or optical flow prediction and visual odometry simultaneously, sub-models are typically used in an unsupervised manner. These models, however, are not effectively integrated, which results in a high number of parameters, which increases the memory needs and computational complexity of the system. The neural model needs to be better integrated, and this is a research topic worth pursuing in the future.

With a DL model, we may acquire several features at once, such as semantics, optical flow features as well as depth information. Different aspects are obtained and matched simultaneously during the encoding stage; they are decoded independently to meet the requirements of the applications during the decoding step.

D. DYNAMIC OBJECTS AND OCCLUSION PROBLEMS

In order to create realistic scenes, developers must consider a range of aspects, such as a large group of moving parts, occlusions, shifting lighting, and varying weather. Most existing facial depth estimation algorithms, on the other hand, simply take into account ideal circumstances. Researchers have made progress in recent years in dealing with moving objects and occlusion environments, but the challenge of accurately estimating the facial depth of complicated environments to satisfy real-world applications remains a major challenge.

E. DATASETS CONSTRUCTION

The consistency and generalization of a learning algorithm are heavily influenced by the quality of the datasets used to train it. Facial depth maps can be improved if more data, with greater quality, and more scene types are available. These available datasets for facial depth maps are limited, and the production of a new dataset is time-consuming and costly. Currently, some researchers are using computers to make a larger number of images for depth maps, but the quality is unstable. In the future, researchers will be looking at how to build a dataset for a monocular face depth map that is suitable for DL.

For instance, synthetic human facial data generation can give better ground truth depth information than can be collected in practice, so high-quality training data can be utilized

to produce better single image depth algorithms. Adding real-data samples, enhancing the hyperrealism of the synthetic datasets, and including a larger range of face characteristics, emotions, and scene illumination could allow for further progress beyond SoA.

VIII. CONCLUSION

The main contribution of this paper is a new lightweight neural facial depth estimation network based on a single image frame depth map. While this network is compatible with previous SoA facial depth estimation techniques, it is substantially smaller in size and computation cost, making it suited for embedded devices and edge-AI applications. When evaluated over four publicly available datasets, this model outperforms SoA on most of the primary measures including RMSE, SqRel and δ_2 . Furthermore, comprehensive experiments show the proposed network's usefulness and generalizability.

A crucial aspect of this research is that training neural facial depth networks on synthetic human facial data produces higher-quality depth maps than is possible through the available realistic datasets. Using lightweight neural single-image depth predictions, high-quality training data may be used to generate accurate facial depth maps. More optimizations beyond SoA should be possible through the incorporation of large and diverse facial depth datasets. Obviously, synthetic facial data will lack the richness of real image datasets of skin features. However, considering the numerous benefits of training a neural depth model with synthetic data, a critical research question is whether it is possible to accomplish comparable results that are answered to SoA facial depth estimation models trained on real-world data. It is possible that future research can include investigation and prosecution into high-resolution facial depth maps, system integration and optimization, high-resolution facial depth map efficiency, data augmentation methods and analyses with a wider range of sample datasets. It would be interesting to investigate a combined multi-tasks network to specifically address downstream applications including image classification, depth maps prediction and semantic segmentation. Another potential future study dimension is multi-frame facial depth, which leverages a succession of image frames and may be paired with some motion estimation or disparity information.

REFERENCES

- [1] Xiong, R., Zhang, S., Gan, Z., Qi, Z., Liu, M., Xu, X., Wang, Q., Zhang, J., Li, F., Chen, X., 2022. A novel 3d-vision-based collaborative robot as a scope holding system for port surgery: a technical feasibility study. *Neurosurgical focus* 52, E13
- [2] Li, Menghan, Bin Huang, and Guohui Tian. "A comprehensive survey on 3D face recognition methods." *Engineering Applications of Artificial Intelligence* 110 (2022): 104669.
- [3] Mertan, A., Duff, D.J., Unal, G., 2022. Single image depth estimation: An overview. *Digital Signal Processing*, 103441
- [4] 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* 31, 4381–4393.
- [5] Attention, V.G.V.A., . Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer .

- [6] Kim, D., Ga, W., Ahn, P., Joo, D., Chun, S., Kim, J., 2022. Global-local path networks for monocular depth estimation with vertical cutdepth. arXiv preprint arXiv:2201.07436 .
- [7] Bhat, S.F., Alhashim, I., Wonka, P., 2021. Adabins: Depth estimation using adaptive bins, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4009–4018.
- [8] Khan, F., Farooq, M.A., Shariff, W., Basak, S., Corcoran, P., 2022. Towards monocular neural facial depth estimation: Past, present, and future. IEEE Access
- [9] Zhang, F., Liu, N., Hu, Y., Duan, F., 2022. Mffnet: Single facial depth map refinement using multi-level feature fusion. Signal Processing: Image Communication 103, 116649.
- [10] Katrolia, J.S., Mirbach, B., El-Sherif, A., Feld, H., Rambach, J., Stricker, D., 2021. Ticam: A time-of-flight in-car cabin monitoring dataset. arXiv preprint arXiv:2103.11719 .
- [11] Eigen, D., Puhrsch, C., Fergus, R., 2014. Depth map prediction from a single image using a multi-scale deep network. Advances in neural information processing systems 27.
- [12] 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.
- [13] Huynh, L., Nguyen-Ha, P., Matas, J., Rahtu, E., Heikkilä, J., 2020. Guiding monocular depth estimation using depth-attention volume, in: European Conference on Computer Vision, Springer, pp. 581–597
- [14] Lee, J.H., Han, M.K., Ko, D.W., Suh, I.H., 2019a. From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326
- [15] Ranftl, R., Bochkovskiy, A., Koltun, V., 2021. Vision transformers for dense prediction, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12179–12188.
- [16] Dovgird, R., Basri, R., 2004. Statistical symmetric shape from shading for 3d structure recovery of faces, in: European conference on computer vision, Springer, pp. 99–113.
- [17] Smith, W.A., Hancock, E.R., 2006. Recovering facial shape using a statistical model of surface normal direction. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1914–1930.
- [18] Zhao, W.Y., Chellappa, R., 2001. Symmetric shape-from-shading using self-ratio image. International Journal of Computer Vision 45, 55–75.
- [19] Jin, Q., Zhao, J., Zhang, Y., 2012. Facial feature extraction with a depth aam algorithm, in: 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, IEEE, pp. 1792–1796.
- [20] Jordan, Caleb. Feature extraction from depth maps for object recognition. Tech. Rep, 2013.
- [21] Arslan, A.T., Seke, E., 2019. Face depth estimation with conditional generative adversarial networks. IEEE Access 7, 23222–23231.
- [22] Kong, D., Yang, Y., Liu, Y.X., Li, M., Jia, H., 2016. Effective 3d face depth estimation from a single 2d face image, in: 2016 16th International Symposium on Communications and Information Technologies (ISCIT), IEEE, pp. 221–230.
- [23] Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y., 2016. Convolutional pose machines, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 4724–4732.
- [24] Yu, F., Wang, D., Shelhamer, E., Darrell, T., 2018. Deep layer aggregation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2403–2412.
- [25] Wang, S., Cheng, Z., Deng, X., Chang, L., Duan, F., Lu, K., 2020. Leveraging 3d blendshape for facial expression recognition using cnn. Sci. China Inf. Sci 63, 1–120114
- [26] Cui, J., Zhang, H., Han, H., Shan, S., Chen, X., 2018. Improving 2d face recognition via discriminative face depth estimation, in: 2018 International Conference on Biometrics (ICB), IEEE, pp. 140–147.
- [27] Moniz, J.R.A., Beckham, C., Rajotte, S., Honari, S., Pal, C., 2018. Unsupervised depth estimation, 3d face rotation and replacement. Advances in neural information processing systems 31.
- [28] Baby, A.T., Andrews, A., Dinesh, A., Joseph, A., Anjusree, V., 2020. Face depth estimation and 3d reconstruction, in: 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA), IEEE, pp. 125–132.
- [29] Chiu, M.T., Cheng, H.Y., Wang, C.Y., Lai, S.H., 2021. High-accuracy rgb-d face recognition via segmentation-aware face depth estimation and mask-guided attention network, in: 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), IEEE, pp. 1–8.
- [30] Chen, J., Niu, S., Gao, X., Li, S., Dong, J., 2022. Sa-unet for face anti-spoofing with depth estimation, in: International Conference on Graphics and Image Processing (ICGIP 2021), p. 16.
- [31] Khan, F., Basak, S., Javidnia, H., Schukat, M., Corcoran, P., 2020. High-accuracy facial depth models derived from 3d synthetic data, in: 2020 31st Irish Signals and Systems Conference (ISSC), IEEE, pp. 1–5.
- [32] Khan, F., Basak, S., Corcoran, P., 2021a. Accurate 2d facial depth models derived from a 3d synthetic dataset, in: 2021 IEEE International Conference on Consumer Electronics (ICCE), IEEE, pp. 1–6.
- [33] 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.
- [34] Alhashim, I., Wonka, P., 2018. High quality monocular depth estimation via transfer learning. arXiv preprint arXiv:1812.11941 .
- [35] Lee, J.H., Han, M.K., Ko, D.W., Suh, I.H., 2019b. From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326
- [36] Khan, F., Basak, S., Corcoran, P., 2021b. Accurate 2d facial depth models derived from a 3d synthetic dataset, in: 2021 IEEE International Conference on Consumer Electronics (ICCE), pp. 1–6. doi:10.1109/ICCE50685.2021.9427595.
- [37] Khan, F., Hussain, S., Basak, S., Lemley, J., Corcoran, P., . An efficient encoder-decoder model for portrait depth estimation from single images trained on pixel-accurate synthetic data. Neural networks: the official journal of the International Neural Network Society 142, 479–491.
- [38] Tan, M., Le, Q., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, pp. 6105–6114.
- [39] 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 .
- [40] Borghi, G., Venturelli, M., Vezzani, R., Cucchiara, R., 2017. Poseidon: Face-from-depth for driver pose estimation, pp. 4661–4670.
- [41] Min, R., Kose, N., Dugelay, J.L., 2014. Kinectfacedb: A kinect database for face recognition. IEEE Transactions on Systems, Man, and Cybernetics: Systems 44, 1534–1548.
- [42] Fanelli, G., Weise, T., Gall, J., Van Gool, L., 2011. Real time head pose estimation from consumer depth cameras. Joint pattern recognition symposium , 101–110.
- [43] Zhou, Q.Y., Park, J., Koltun, V., 2018. Open3d: A modern library for 3d data processing. arXiv preprint arXiv:1801.09847 .



3D reconstruction

FAISAL KHAN received the BS in Mathematics from University of Malakand Chankdara lower Dir, Pakistan, in 2015, the M.Phil. degree in Mathematics from the Hazara University Mansehra Pakistan, in 2017. He is currently pursuing the PhD degree with the National University of Ireland Galway (NUIG). He is also with FotoNation/Xperi. His research interest is machine learning using deep neural networks for tasks related to computer vision including Depth estimation and



WASEEM SHARIFF received his B.E degree in computer science from Nagarjuna College of Engineering and Technology (NCET) in 2019 and his M.S. degree in computer science, specializing in artificial intelligence from National University of Ireland Galway (NUIG) in 2020. He is currently working as a research engineer at Xperi, Inc. He is also pursuing his PhD degree (IRC) at the National University of Ireland Galway. His research interests include machine learning for computer vision

applications, with a particular emphasis on automotive in-cabin monitoring applications.



MUHAMMAD ALI FAROOQ received his BE degree in electronic engineering from IQRA University in 2012 and his MS degree in electrical control engineering from the National University of Sciences and Technology (NUST) in 2017. He has completed his PhD degree at the National University of Ireland Galway (NUIG). Currently, he is employed as a Post-Doctoral researcher at the University of Galway. His research interests include machine vision, computer vision, video

analytics, and sensor fusion. He has won the prestigious H2020 European Union (EU) scholarship and currently working as one of the consortium partners in the Helias (thermal vision augmented awareness) project funded by the EU.



SHUBHAJIT BASAK received the B.Tech. degree in Electronics and Communication Engineering from the West Bengal University of Technology, India, in 2011 and M.Sc. in Computer Science from the National University of Ireland Galway, Ireland in 2018. He had more than 6 years of industrial experience as a software development professional. He is currently pursuing the Ph.D. degree in Computer Science with the National University of Ireland Galway, Ireland. He is also

with FotoNation/Xperi. His research interest includes deep learning tasks related to computer vision.



PETER CORCORAN (Fellow, IEEE) holds the Personal Chair in electronic engineering at the College of Science and Engineering, National University of Ireland Galway. He is currently an IEEE Fellow recognized for his contributions to digital camera technologies, notably in-camera red-eye correction and facial detection. He was a CoFounder in several start-up companies, notably FotoNation, now the Imaging Division of Xperi Corporation. He has over 600 technical publica-

tions and patents, over 100 peer-reviewed journal articles, 120 international conference papers, and a coinventor of more than 300 granted U.S. patents. 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.

...

Appendix H

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

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.

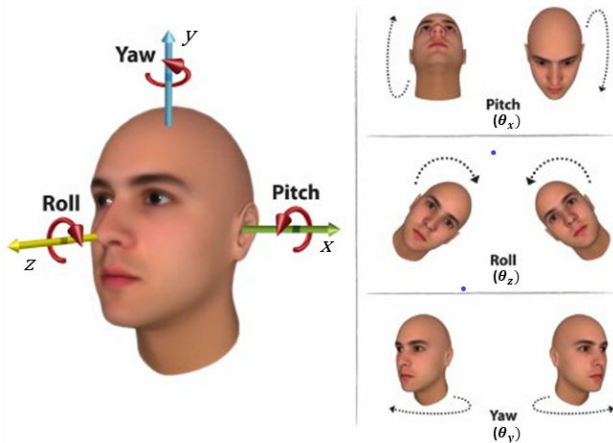


FIGURE 1. Head orientation with Pitch, Yaw and Roll [1].

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

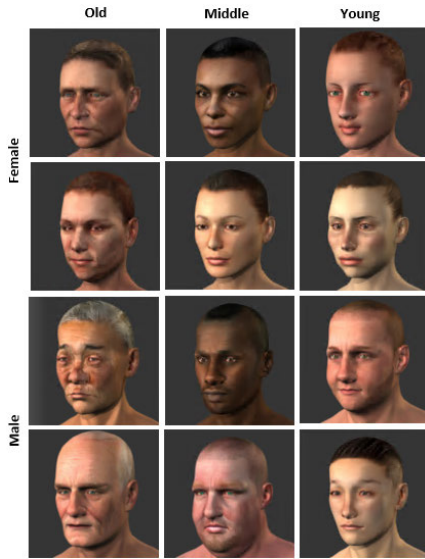


FIGURE 2. Samples from the 100 Realistic Head Models with variation in gender, race and age.

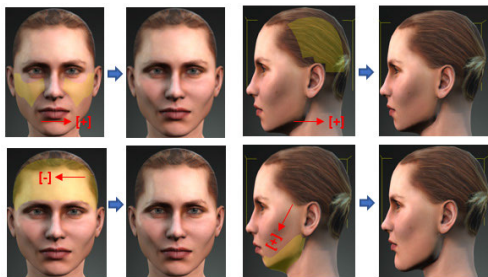


FIGURE 3. Applying change in the morph to add variations in the head models in iClone [45].

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.

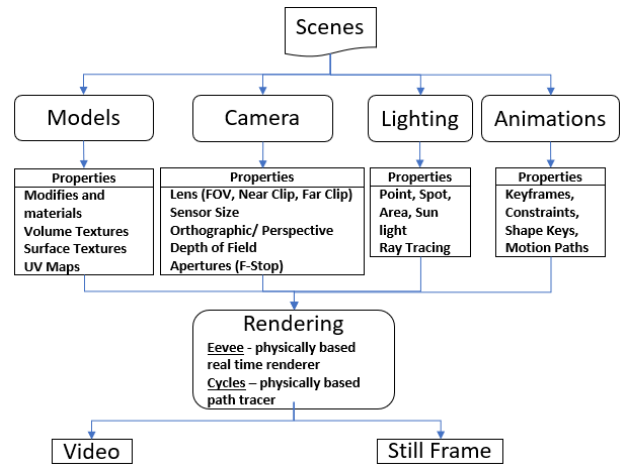


FIGURE 4. Workflow and different features of Blender [45].

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.

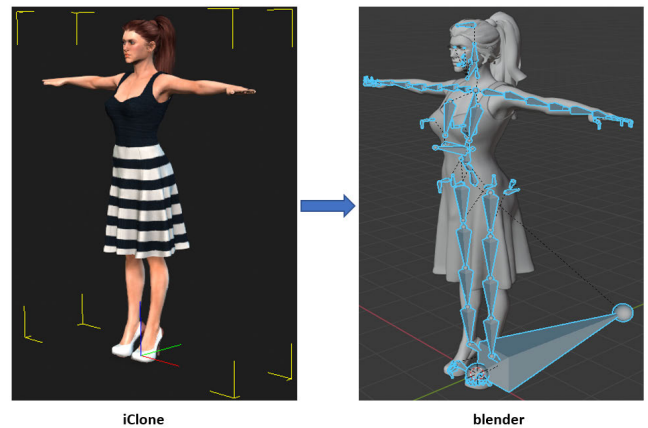


FIGURE 5. Importing the fully rigged FBX models from iClone to Blender [45].

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.

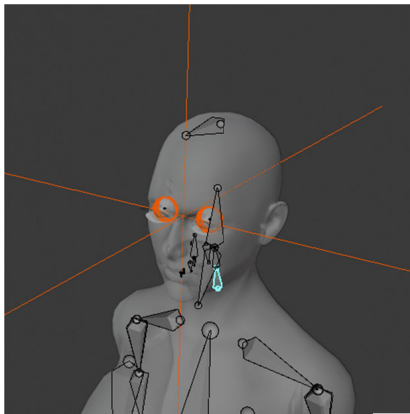


FIGURE 6. Neck bone highlighted in cyan on which the head rotation has been applied and the empty object at the center of the two eyeballs highlighted in orange.

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

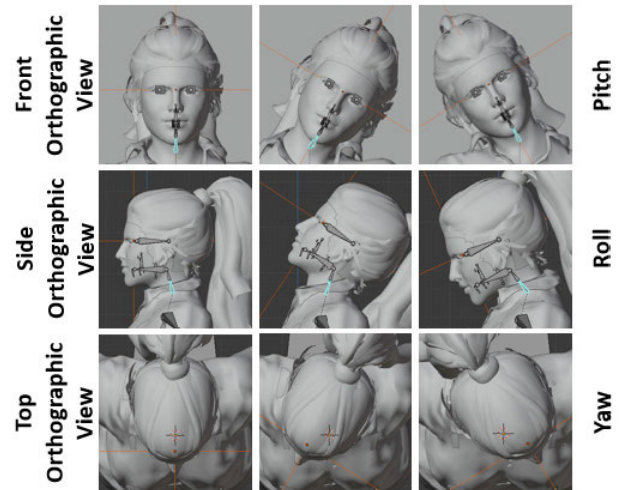


FIGURE 7. Applying head pose along the three axes with respect to the neck bone highlighted.

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>

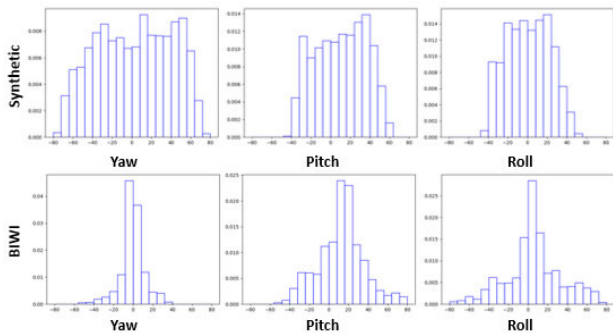


FIGURE 9. The first row shows the label distribution of the generated data across yaw, pitch and roll. The second row shows the similar distribution of Biwi data.

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

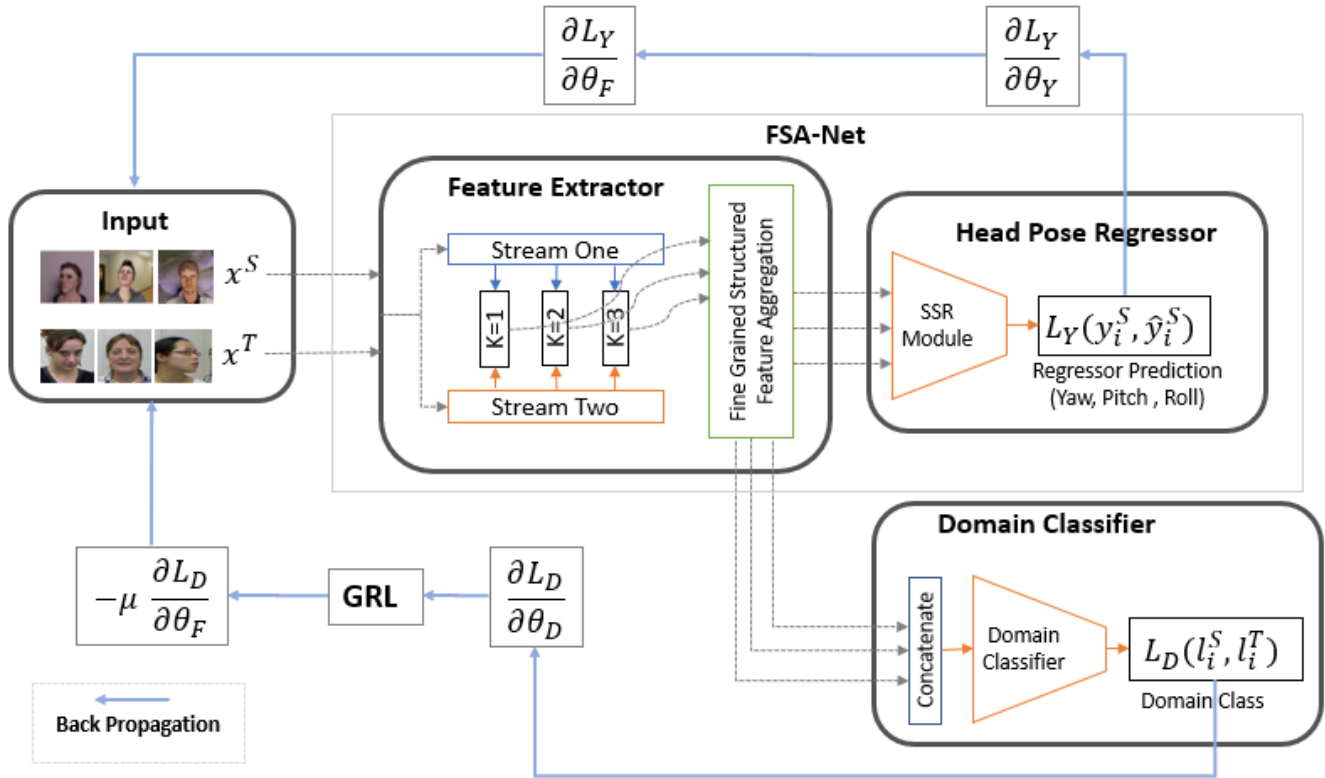


FIGURE 10. FSA-Net with the Domain Classifier and GRL layer for the adversarial learning.

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

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

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

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

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 –

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

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.

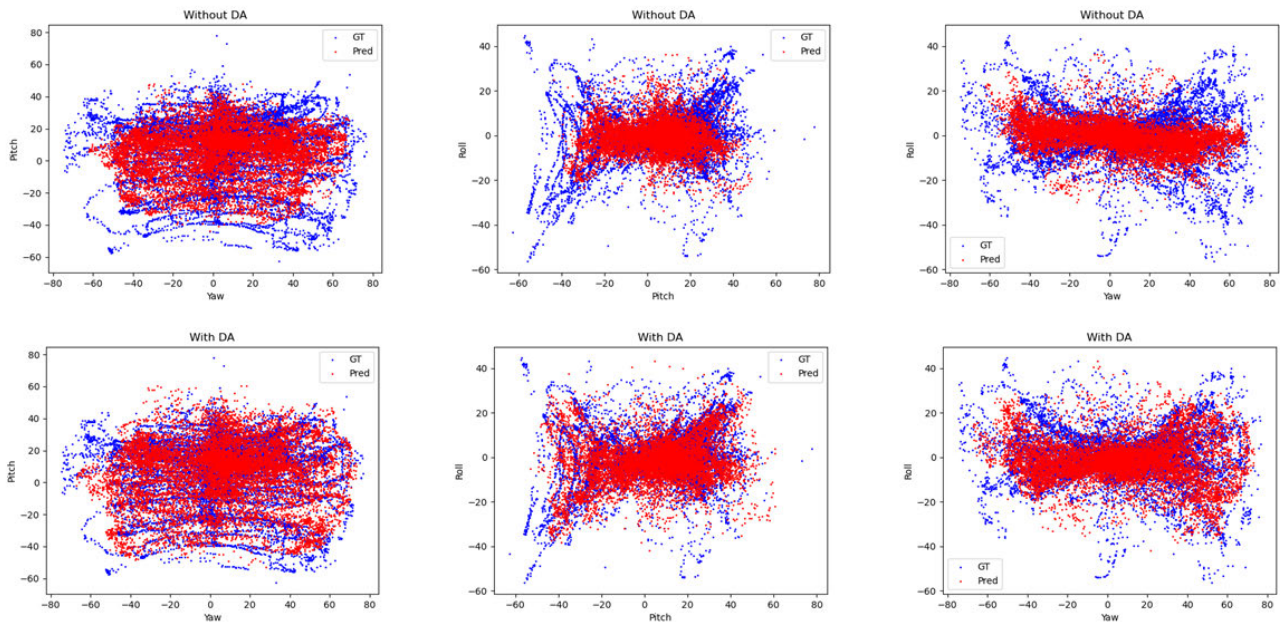


FIGURE 11. Distribution of ground truth and predicted labels in blue and red color respectively. The first row shows the result without domain adaptation and the second row shows with domain adaptation. The first column is Yaw versus Pitch, second column is Pitch versus Roll and the third column shows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

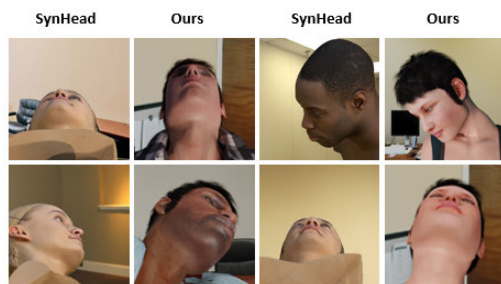


FIGURE 12. Samples from SynHead [7] dataset with artefacts because of large-concatenated rotation angles and samples from the dataset produced from this work with similar head rotations.

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.

• • •

Appendix I

Methodology for Building Synthetic Datasets with Virtual Humans

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

This work is funded by Science Foundation Ireland Centre for Research Training in Digitally Enhanced Reality (D-REAL) under grant 18/CRT/6224.

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

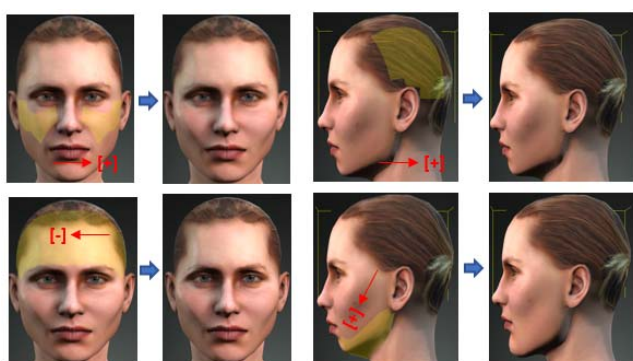


Fig. 2. Adjust cheek, forehead, skull and chin bones in Character Creator

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).

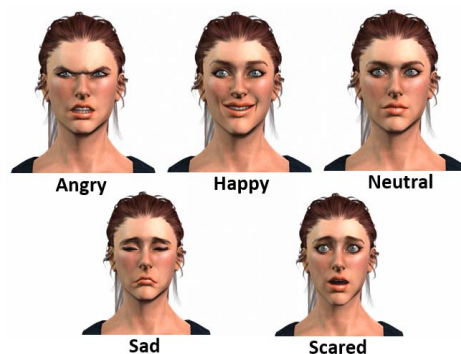


Fig. 3. Sample images with different expression rendered from iClone

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.

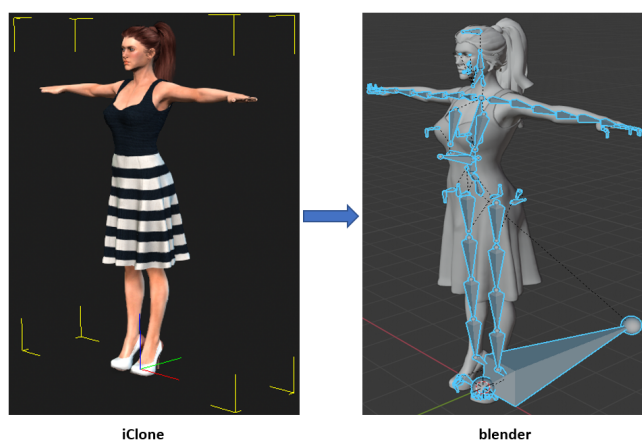


Fig. 4. Sample of a fully rigged model imported in Blender from iClone

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.

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.

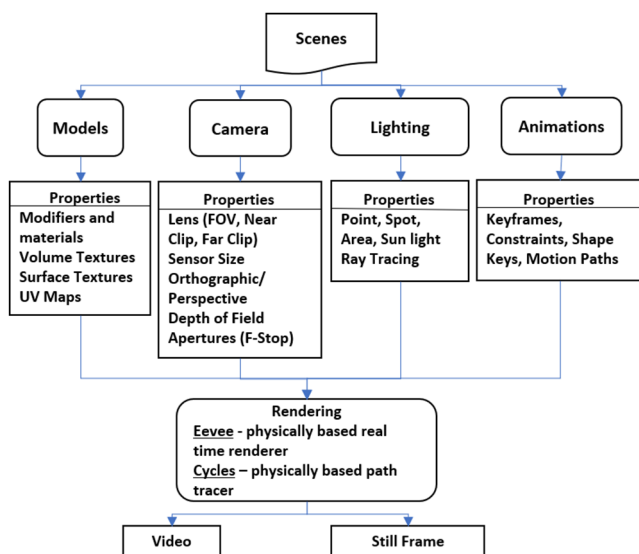


Fig. 5. Sample workflow in Blender

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.

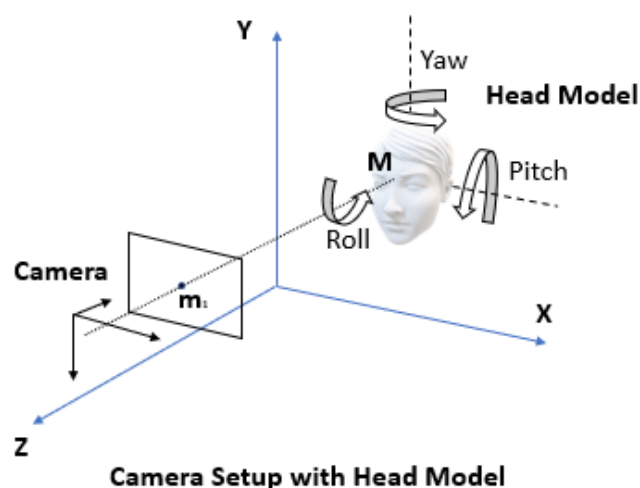


Fig. 6. Sample setup of camera and the model

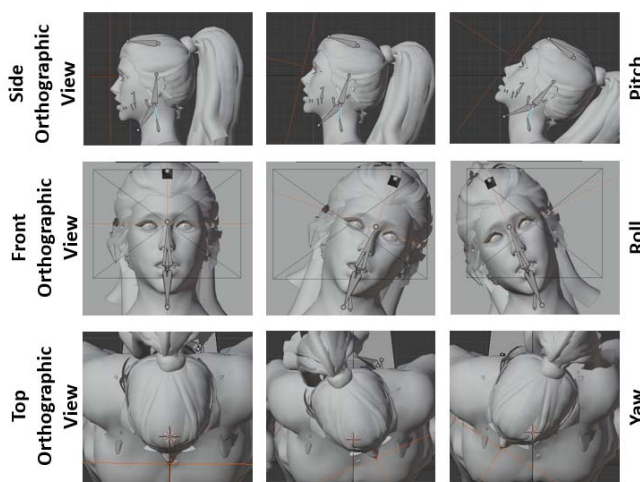


Fig. 7. Applying head movement (yaw, roll, and pitch) on the model in Blender to capture the head pose

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.

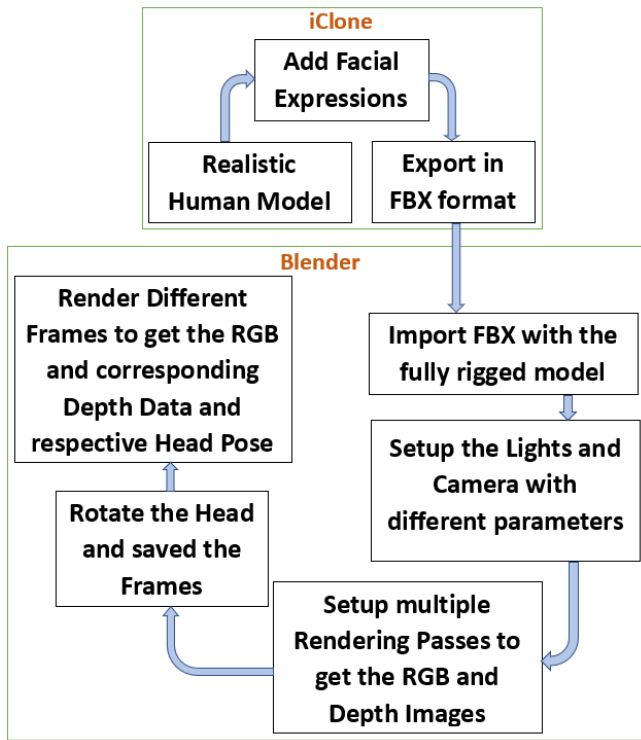


Fig. 8. Pipeline to produce a virtual human

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

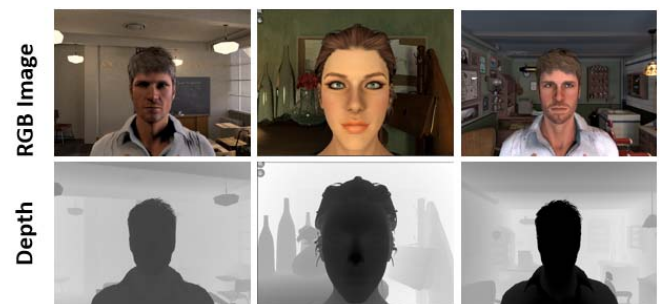






Fig. 11. Sample images and their depth image (scaled to visualize) with more complex background

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.

Appendix J

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

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

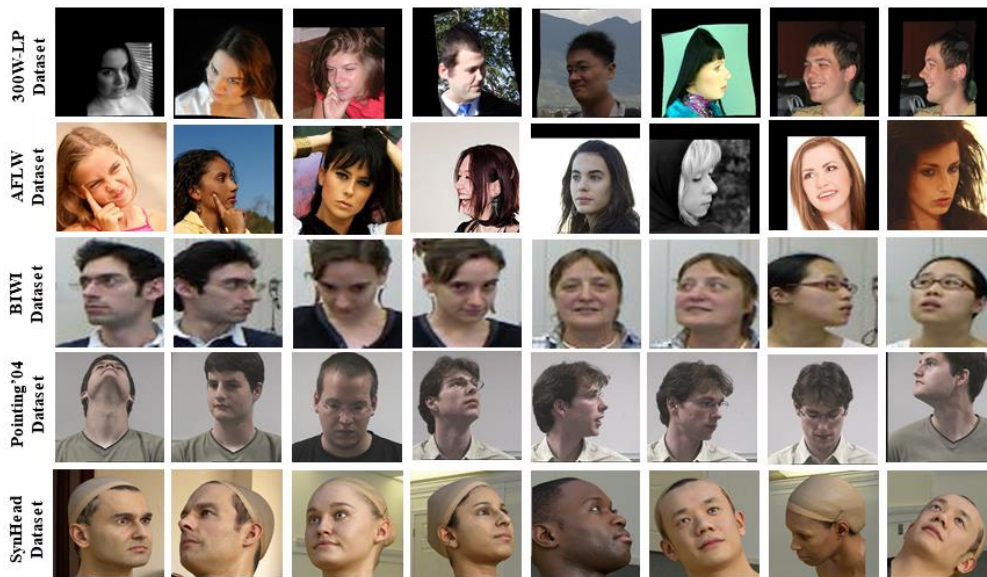


Figure 1. Sample Images from different datasets

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

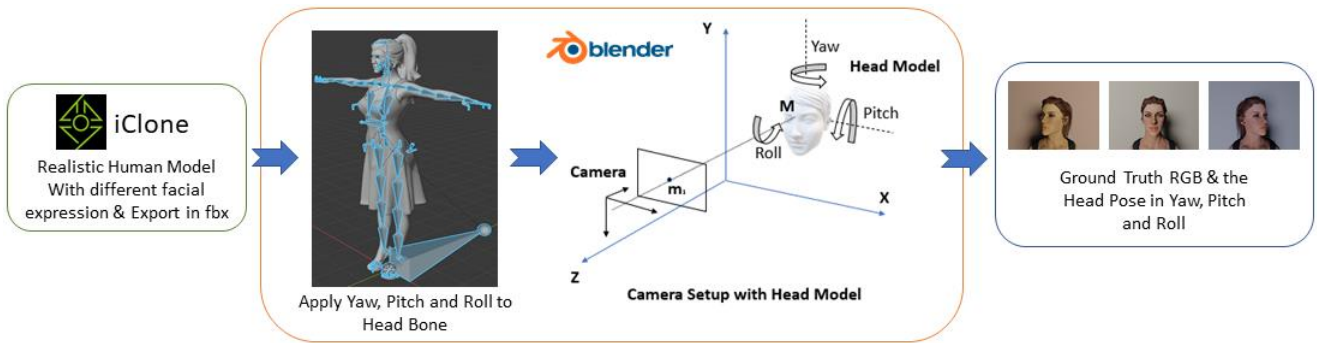


Figure 2. Overall Pipeline to produce the synthetic Head Pose Data

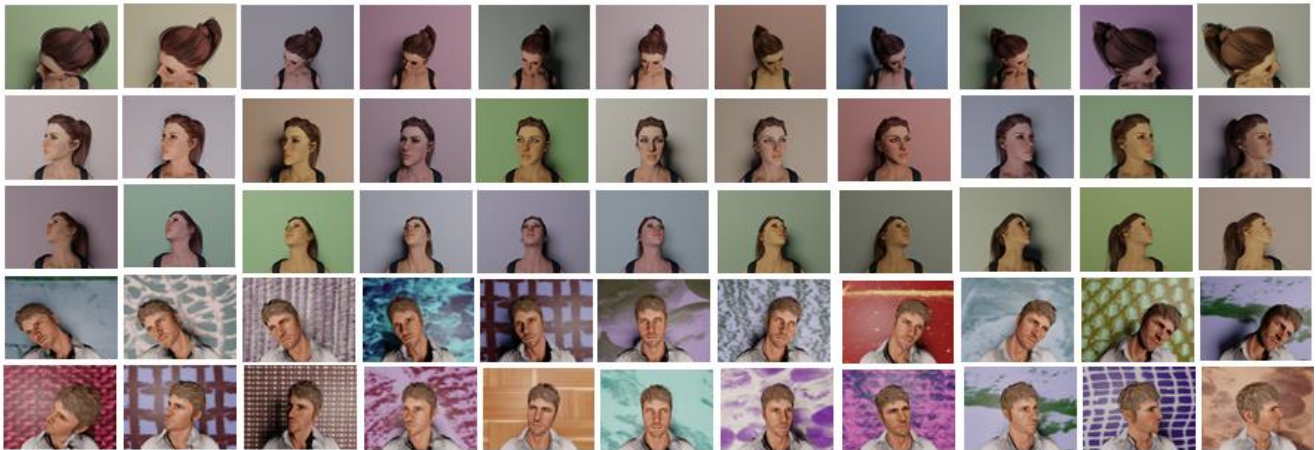


Figure 3. Samples from our dataset with plain and textured background and varying Yaw, Pitch, and Roll

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.

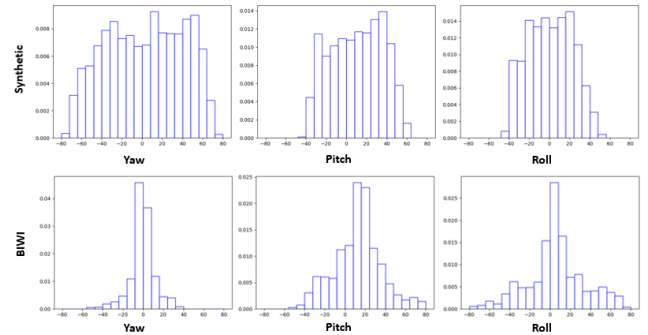


Figure 4. The first row shows the data distribution of Yaw, Pitch and Roll in our synthetic dataset and the second row shows the same distribution from Biwi Test dataset

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).