National University of Ireland Galway

Doctoral Thesis

Detecting Seen/Unseen Objects with Reducing Response Time for Multimedia Event Processing

Author:
Asra Aslam

Supervisor:
Dr. Edward Curry

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Insight Centre for Data Analytics
National University of Ireland Galway

August 2021

Declaration of Authorship

I, Asra Aslam, declare that this thesis titled, 'Detecting Seen/Unseen Objects with Reducing Response Time for Multimedia Event Processing' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

NATIONAL UNIVERSITY OF IRELAND, GALWAY

Abstract

Insight Centre for Data Analytics
National University of Ireland Galway

Doctor of Philosophy

Detecting Seen/Unseen Objects with Reducing Response Time for Multimedia Event Processing

by Asra Aslam

The enormous growth of multimedia content in the Internet of Things (IoT) domain leads to the challenge of processing multimedia streams in real-time; thus, the Internet of Multimedia Things (IoMT) is an emerging concept in the field of smart cities. In the current scenario, we expect that real-time image processing systems are robust in performance, but they are designed for specific domains like traffic management, security, parking, supervision activities, etc. Existing event-based systems are designed to process event streams according to user subscriptions and focused only on structured events like energy consumption events, RFID tag readings, finance, packet loss events, etc. However, multimedia content occupies a significant share in IoT compared to the scalar data obtained from conventional IoT devices. Due to the lack of support for processing multimedia events in existing event-based systems of IoT, there is the need for an Internet of Multimedia Things (IoMT) based event processing system which can also process images/videos.

Multiple applications within smart cities may require the processing of numerous seen and unseen concepts (unbounded vocabulary) in the form of subscriptions. Deep neural network-based techniques are effective for image recognition, but the limitation of having to train classifiers for unseen concepts may increase the overall response-time for multimedia-based event processing models. These models require a massive amount of annotated training data (i.e., images with bounding boxes). It is not practical to have all trained classifiers or annotated training data available for a large number of unseen classes of smart cities. In this thesis, I address the problem of training classifiers online for unseen concepts to answer user queries that include processing multimedia events in minimum response time and maximum accuracy for the IoMT based systems.

The contributions of this thesis are manifold. First, I analyze the trends, challenges, and opportunities in the state-of-the-art for the IoMT based systems. I propose a generalizable event processing approach to consume IoMT data as a native event type and optimize it for different scenarios consisting of seen, unseen, and partially unseen concepts. The first domain-specific classifier-based model enables the feature extraction in event processing based on subscriptions and optimizes the *testing time* using an elementary *classifier division and selection* approach. Next, I propose the *hyperparameters* based multimedia event detection model to handle completely unseen concepts and optimize the training time for the training from scratch. However, for the partially unseen concepts, I propose a *domain adaptation* based model that enables knowledge transfer from seen to unseen (like bus $\rightarrow$ car) concepts and reduces classifiers' overall response time. The final specific model handles the challenge of collecting a large number of images with bounding box annotations for the training of object detection models on unseen concepts. In this model, I propose a detector (named UnseenNet) to train unseen classes using only image-level labels with no bounding boxes annotation.

I primarily include You Only Look Once (YOLO), Single Shot MultiBox Detector (SSD), and RetinaNet for the object detection while having seen/unseen classes belongs to Pascal VOC, Microsoft COCO, and OpenImages detection datasets. The results indicate that the proposed multimedia event processing models achieve accuracy of 66.34% within 2 hours using classifier division and selection approach, 84.28% within 1 hour using hyperparameter-based optimization, and 95.14% using domain adaptation-based optimization within 30 min of response-time on real-time multimedia events. Lastly, evaluations of domain adaptation based model without bounding boxes demonstrate that UnseenNet outperforms the baseline approaches and reduces the training time of days or $>$5.5 hours to $<$5 minutes.

Dedicated to Ammi, Papa, Samar, and Alia...

Acknowledgements

Thanks, is a small word for the gratitude I want to communicate,
For the ones who are happy more than me, my lines can never enumerate.
Let us give it a trial to make these acknowledgments worthwhile!
No doubt, an incredible thanks goes to the almighty
to bless me with supportive people wisely.

Words like respect, thanks, gratitude, praise start to become tiny & blurry,
In front of the sincere gratitude that I want to express to my advisor Dr. Edward Curry
None of the expressions can justify the time and dedication that he made me received,
Countless feedbacks, endless support, and still being patient, difficult to be believed.

His priceless guidance, continuous advice, in all good/bad time of research and writing,
I am definitely failing to communicate using these conventional words of citing.

Thanks a million may be significantly less to say for Fabiana and Aisling
The way they filled the first and second half of Ph.D. with beautiful spring
Special thanks to amazing friends Nishma, Daniela, Thu, Safina, Hugo, & others
Who supported me on this journey can't count many more on my fingers.

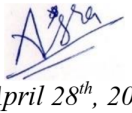
An exceptional thanks to Niki and Atiya, to make this road trip fantastic
Thank you, Tarek, Felipe, and everyone who made this highway enthusiastic

Warm thanks to my mummy and papa for their guidance, effort, and sacrifice
My tiny thanks cannot pay back any price
To say one truth, I will never feel shy
Papa, you were my hardest Goodbye

I am very grateful to my husband Samar, my friend, my companion
Your existence beside me made every moment a beautiful one.
Alia my daughter, you are the most wonderful gift I ever got
Your laugh, your trust, your energy makes me smile a lot.

In any case, how can I forget to say thanks to Science Foundation Ireland (SFI),
And the European Regional Development Fund for funding this work of mine.

I wish I could write more, but this is the thesis shore!



April 28th, 2021
(1:45 am)

Contents

Declaration of Authorship i

Abstract ii

Acknowledgements v

Contents vi

List of Figures xii

List of Tables xiv

1 Introduction 1
- 1.1 Introduction . . . 1
- 1.2 Problem Overview and Motivation . . . 2
- 1.3 Motivational Scenario . . . 4
- 1.4 Problem Statement . . . 4
- 1.5 Proposed Approach . . . 6
  - 1.5.1 Domain-Specific Classifier based Multimedia Event Detection . . . 8
  - 1.5.2 Hyper-Parameters based Adaptive Multimedia Event Detection . . . 9
  - 1.5.3 Domain Adaptation based Multimedia Event Detection . . . 9
  - 1.5.4 Domain Adaptation based Multimedia Event Detection without Bounding Boxes . . . 10
- 1.6 Research Hypothesis . . . 10
- 1.7 Research Methodology . . . 11
- 1.8 Core Contributions . . . 12
- 1.9 Thesis Organization . . . 13
- 1.10 Summary of Conclusions . . . 15
- 1.11 Associated Publications . . . 15

2 Problem Formulation 17
- 2.1 Introduction . . . 17
- 2.2 Problem Domain and Technical Limitations . . . 18
- 2.3 Motivation . . . 20
  - 2.3.1 Scenarios . . . 20

2.3.2 Generalizability . . . 21
2.4 Requirements . . . 22
2.5 Challenges and Opportunities . . . 23
2.5.1 Challenges . . . 23
2.5.2 Opportunities . . . 24
2.6 Response Time Problem Formulation . . . 25
2.7 Research Questions . . . 27
2.8 Summary . . . 28

3 Background and Related Work **29**
3.1 Introduction . . . 29
3.2 Multimedia in Internet of Things . . . 29
3.2.1 Concepts of IoT & IoMT . . . 30
3.2.1.1 Internet of Things . . . 30
3.2.1.2 Internet of Multimedia Things . . . 31
3.2.2 Characteristics, Requirements, and Solutions for IoMT . . . 33
3.3 Object Detection . . . 37
3.3.1 Deep Neural Network based Object Detection Models . . . 37
Faster R-CNN . . . 37
Single Shot Detection (SSD) . . . 38
You Only Look Once (YOLO) . . . 40
RetinaNet . . . 41
Comparison of DNN based Object Detection Models: . . . 41
Deep Neural Network based Object Detection Models: . . . 43
3.3.2 Seen Classes based Object Detection Datasets . . . 43
Image_Net: . . . 43
Pascal VOC: . . . 43
Microsoft COCO: . . . 44
Open Images Dataset . . . 45
Comparison of Object Detection based Datasets . . . 46
3.3.3 Small Datasets based N-Shot learning methods . . . 48
Zero-Shot . . . 48
One-Shot . . . 49
Few-Shot . . . 49
3.4 Summary . . . 49

4 Seen/Unseen Objects based Multimedia Event Processing **50**
4.1 Introduction . . . 50
4.2 Redefining Event Processing to Multimedia Event Processing . . . 51
4.2.1 Defining Generalizability . . . 52
4.2.2 Detect Operator . . . 53
4.2.3 Unseen Subscriptions . . . 54
4.3 Scenarios for Unseen Subscriptions of Multimedia Event Processing . . . 54
4.4 Adaptive Classifier based Multimedia Event Processing Models . . . 57
4.4.1 Domain-Specific Classifier based Multimedia Event Detection Model 58
4.4.2 Hyper-Parameters based Adaptive Multimedia Event Detection Model . . . 59

4.4.3 Domain Adaptation based Multimedia Event Detection Model . . 60
4.4.4 Domain Adaptation based Multimedia Event Detection Model without Bounding Boxes . . . 60
4.4.5 Deployment of Multimedia Event Processing Models . . . 61
4.5 Discussion . . . 62
4.6 Summary . . . 64

5 Domain-Specific Classifier based Multimedia Event Detection 66
5.1 Introduction . . . 66
5.2 Problem Overview . . . 67
5.2.1 Preliminaries . . . 67
5.2.2 Motivational Scenarios . . . 68
5.2.3 Problem Statement . . . 69
5.3 Background and Related Work . . . 69
5.3.1 Event-based Approaches for IoT . . . 69
5.3.2 Application-Specific Approaches for IoMT . . . 73
5.3.3 Multimedia Query Languages . . . 76
5.3.4 Gap Analysis . . . 77
5.4 Proposed Approach . . . 80
5.5 Designing and Implementation . . . 81
5.5.1 Generalized Multimedia Event Processing Engine . . . 81
5.5.1.1 Receiver . . . 81
5.5.1.2 Multimedia Event Processing Engine (MEPE) . . . 81
Matcher: . . . 81
Multimedia Event Processing Language (MEPL): . . . 81
Subscription Covering based Optimization: . . . 81
Feature extraction: . . . 82
Classifiers: . . . 82
5.5.1.3 Forwarder . . . 83
5.5.2 Multimedia Event Processing Algorithms . . . 84
5.5.2.1 Multimedia Event Processing Engine . . . 84
5.5.2.2 Subscription Covering based Optimization . . . 85
5.6 Evaluation . . . 86
5.6.1 Evaluation Methodology . . . 86
5.6.2 Evaluation Metrics . . . 88
5.6.3 Experiments and Results . . . 88
5.6.3.1 Evaluation of Feature Extraction . . . 88
5.6.3.2 Proof of Optimization . . . 91
Throughput vs Number of Classes . . . 91
Precision-Recall vs N-Class Classifiers . . . 91
5.7 Conclusion and Discussion . . . 93

6 Hyper-Parameters based Adaptive Multimedia Event Detection 95
6.1 Introduction . . . 95
6.2 Problem Overview . . . 97
6.2.1 Preliminaries . . . 97
6.2.2 Motivational Scenarios . . . 98

6.2.3 Problem Statement . . . 99
6.3 Background and Related Work . . . 99
6.3.1 Online Learning of Classifiers . . . 99
6.3.2 Self-Tuning of Classifiers . . . 101
6.3.3 Gap Analysis . . . 103
6.4 Proposed Approach . . . 103
6.5 Designing and Implementation . . . 104
6.5.1 Adaptive Hyper-Parameter based Multimedia Event Processing Engine . . . 104
6.5.2 Adaptive Hyper-Parameter based Multimedia Event Processing Algorithms . . . 110
6.6 Evaluation . . . 112
6.6.1 Evaluation Methodology . . . 112
6.6.1.1 Strategies . . . 113
Minimum Response Time needed while Minimum Accuracy allowed: . . . 113
Optimal Response Time needed while Optimal Accuracy allowed: . . . 113
Maximum Response Time allowed while Maximum Accuracy needed: . . . 114
6.6.2 Evaluation Metrics . . . 114
6.6.3 Experiments and Results . . . 115
6.6.3.1 Online Classifier Construction before Adaptation . . . 115
Response Time vs Performance of Object Detection Models before Adaptation . . . 115
Results for Proposed Strategies on Selected Object Detection Model . . . 117
6.6.3.2 Online Classifier Construction after Adaptation . . . 118
Hyperparameter Tuning . . . 118
Response Time vs Performance of Object Detection Models after Adaptation . . . 121
Response-Time Driven Precision-Recall Area Under Curve (AUC) . . . 123
Results for Proposed Strategies on Selected Object Detection Model . . . 124
6.7 Conclusion and Discussion . . . 126

7 Domain Adaptation based Multimedia Event Detection 128
7.1 Introduction . . . 128
7.2 Problem Overview . . . 129
7.2.1 Preliminaries . . . 129
7.2.2 Motivational Scenarios . . . 130
7.2.3 Problem Statement . . . 131
7.3 Background and Related Work . . . 131
7.3.1 Knowledge Transfer from Training to Testing Data Distribution . 132
7.3.2 Knowledge Transfer for Object Detection Domain Shift Distribution 133
7.3.3 Gap Analysis . . . 134
7.4 Proposed Approach . . . 134

7.5 Designing and Implementation . . . 135
7.5.1 Transfer Learning based Domain Adaptive Multimedia Event Processing Engine . . . 135
7.5.2 An Approach for Domain Adaptation . . . 137
7.5.3 Domain Adaptation based Multimedia Event Processing Algorithms 139
7.6 Evaluation . . . 140
7.6.1 Evaluation Methodology . . . 140
7.6.2 Evaluation Metrics . . . 140
7.6.3 Experiments and Results . . . 141
7.6.3.1 Performance–Response-Time Trade-off of Object Detection Models . . . 141
7.6.3.2 Empirical Analysis for Domain Shift . . . 144
7.6.3.3 Simulation on Proposed Model . . . 146
Results on Domain Adaptation . . . 146
Confusion Matrix . . . 147
Seen/Unseen Domains . . . 147
7.7 Conclusions and Discussion . . . 148

8 Domain Adaptation based Multimedia Event Detection without Bounding Boxes 151
8.1 Introduction . . . 151
8.2 Problem Overview . . . 152
8.2.1 Preliminaries . . . 153
8.2.2 Motivational Scenarios . . . 154
8.2.3 Problem Statement . . . 155
8.3 Background and Related Work . . . 155
8.3.1 Weakly Supervised Object Detection (WSOD) with Knowledge Transfer . . . 155
8.3.2 Large Scale Detection through Adaptation (LSDA) . . . 156
8.3.3 Gap Analysis . . . 157
8.4 Proposed Approach . . . 159
8.4.1 Baseline LSDA . . . 159
8.4.2 UnseenNet: Designing and Implementation . . . 161
8.4.2.1 Training Baseline Detector Offline for Seen Concept (with Bounded Vocabulary) . . . 162
8.4.2.2 Training Online Detector for Unseen Concept (for Unbounded Vocabulary) . . . 163
8.4.2.3 Implementation Details . . . 164
Data Preparation . . . 164
Training . . . 165
Degree of Similarity Parameter (α) . . . 166
Estimation of Number of Epochs . . . 166
8.5 Evaluation . . . 167
8.5.1 Evaluation Methodology . . . 167
8.5.2 Evaluation Metrics . . . 167
8.5.3 Experiments and Results . . . 167
8.5.3.1 Quantitative Evaluation on Unseen Categories . . . 167

Comparative Analysis with Existing Models 167
Experimental Results with Response-Time 168
Experimental Results with Unseen Concepts 171
8.5.3.2 Qualitative Evaluation on Unseen Categories 171
8.6 Conclusion and Discussion 173

9 Conclusion and Future Work **175**
9.1 Thesis Summary 175
9.2 Conclusions 177
9.3 Core Contributions 180
Neural Network based Matcher with "DETECT" Operator: 180
Standardization of objective function "Response-Time": . . 181
Adaptive Framework for Online Classifier Construction: . . 181
Instantiation of Online Classifier Learning model using Fine-tuning & Freezing Neural-Network Layers: . . . 181
Evaluation of Proposed Models using Object Detection methods with Response-Time & Accuracy: 181
UnseenNet: LSDA based Detector with Online Training using only Image-Level labels: 181
Derivation of Minimum and Maximum limits of Response-Time for Weakly Supervised Learning: 182
9.4 Limitations & Open Questions 182
9.5 Future Research Directions 183

Bibliography **186**

List of Figures

1.1 Global Internet Traffic by Application Category [1] . . . 2
1.2 Generalizable Multimedia Event Processing . . . 4
1.3 Response Time in Multimedia Event Processing . . . 5
1.4 Scenarios for Multimedia Event Processing adhering to Seen/Unseen Concept Problem . . . 7

2.1 Problem analysis of Multimedia Event Processing using Online Training of Classifiers for Seen/Unseen Objects . . . 19
2.2 Multimedia Processing Events Scenario . . . 21
2.3 Definition of Response-Time . . . 26

3.1 Visions of IoT [2] to our visions of IoMT . . . 32
3.2 Neural Network based Architectures of Object Detection Models. . . . 39
3.3 Example Images for Object Detection Datasets . . . 44
3.4 Number of Images Annotated in Object Detection Datasets with classes. . 46

4.1 Current Approaches for Multimedia Event Processing . . . 51
4.2 IoMT Aware Middleware . . . 52
4.3 Scenarios for Adaptive Multimedia Event Processing adhering to Seen/Unseen Concept Problem . . . 55
4.4 Adaptive Multimedia Event Processing with Proposed Models I: Domain-Specific Classifier, II: Hyper-Parameters, III: Domain Adaptation, IV: Domain Adaptation without Bounding Boxes based Multimedia Event Detection. . . . 56
4.5 Summarizing Proposed Techniques for Adaptive Multimedia Event Processing . . . 65

5.1 Multimedia Event Processing in Existing Domain Specific Applications . . 68
5.2 Scenarios for Multimedia Event Processing adhering to Seen/Unseen Concept Problem . . . 69
5.3 IoMT based Multimedia Event Processing model . . . 82
5.4 System Optimization using Subscriptions (Seen/Unseen Concepts) . . . 83
5.5 Detect Operator on Traffic, Sports, Home and Animal related events . . . 91
5.6 Average Throughput of Classifiers . . . 92
5.7 Average Precision-Recall of Classifiers . . . 93

6.1 Conceptual Architecture for the Online Adaptive Classifier based Multimedia Event Processing . . . 96
6.2 Scenarios for Multimedia Event Processing adhering to Seen/Unseen Concept Problem . . . 98

6.3 Scenario-1: Completely Unseen Concept Arrived (Application of Model-II: Hyper-Parameters based Adaptive Multimedia Event Detection) 99
6.4 Adaptation Model for Multimedia Event Processing 105
6.5 General Structure Confusion Matrix for Evaluations 116
6.6 Performance vs Response Time without Adaptation (for 15-min and 60-min intervals) 117
6.7 Hyperparameter Tuning for 15-min and 1-hour training 119
6.8 Performance vs Response Time after Adaptation (for 15-min and 60-min intervals) 122
6.9 Area Under Curve (AUC) Before and After proposed Adaptation within Response-Time interval of 15 min and 1 hour using different Object Detection Models 124

7.1 Scenarios for Multimedia Event Processing adhering to Seen/Unseen Concept Problem 130
7.2 Scenario-2: Adaptation Possible from Seen to Unseen Concept with Bounding Boxes (Application of Model-III: Domain Adaptation based Multimedia Event Detection) 131
7.3 Domain Adaptation 131
7.4 Transfer Learning based Domain Adaptive Classifier Construction for Multimedia Event Processing 136
7.5 Techniques used for Transfer Learning 139
7.6 Performance vs Response Time with and without Adaptation 142
7.7 Analysis for Domain Shift 145
7.8 Performance vs Response Time with Domain Adaptation 146
7.9 Response Time with Unseen Subscriptions 147

8.1 Scenarios for Multimedia Event Processing adhering to Seen/Unseen Concept Problem 154
8.2 Annotations with/without Bounding Boxes 155
8.3 Conceptual Representation of LSDA: Large Scale Detection through Adaptation [3] 156
8.4 An illustration of our "UnseenNet" model. 160
8.5 mAP with parameter α for degree of similarity. 166
8.6 Examples of mAP with Response-Time, For each "Unseen" category, we use the top-10 weighted average nearest neighbor "Seen" categories for adaptation. 170
8.7 mAP of our model on 100 "Unseen" Categories within 5 min of training. . 171
8.8 Examples of correct detections of our model on "Unseen" categories are shown in red color and groundtruth (taken from OID) in green. Last two row unseen classes are downloaded online, and no groundtruth available to date. 172
8.9 Examples of Incorrect detections (Label Object Correctly but Incorrect Localization) are shown in red and groundtruth (taken from OID) in green. Last two row unseen classes are downloaded online, and no groundtruth available to date. 173

List of Tables

1.1 Summarizing proposed work with analyzed sub problems of multimedia event processing. . . . 8

3.1 Comparison of IoT and IoMT based systems . . . 34
3.2 Comparison of DNN based Object Detection Models . . . 42
3.3 Comparison of Available Object Detection Datasets . . . 47

5.1 Analysis of Related-Work with identified Requirements . . . 78
5.2 Description of Symbols . . . 85
5.3 Performance of proposed MSPE on different classifiers in terms of Accuracy and Throughput (FPS) . . . 90

6.1 Analysis of Related-Work with identified Requirements . . . 100
6.2 Performance of Existing Object Detection Models . . . 108
6.3 Hyperparameter values for Adaptive Training . . . 112
6.4 Default Hyperparameters with Accuracy for different strategies. . . . 118
6.5 Space defined for Hyperparameter Tuning for the Scratch Training of 15-min and 1-hour. . . . 120
6.6 Derived Hyperparameters with Accuracy for Strategy-1 . . . 126
6.7 Derived Hyperparameters with Accuracy for Strategy-2 . . . 126
6.8 Confusion Matrix for Strategy S1: Minimum Response Time needed while Minimum Accuracy allowed . . . 127
6.9 Confusion Matrix for Strategy S2: Optimal Response Time needed while Optimal Accuracy allowed . . . 127
6.10 Confusion Matrix for Strategy S3: Maximum Response Time allowed while Maximum Accuracy needed . . . 127

7.1 Analysis of Related-Work with identified Requirements for Knowledge Transfer . . . 133
7.2 Evaluations on Domain Adaptation on multiple training techniques . . . 143
7.3 Detection mAP on Specific Domain Transfers using different Domain Adaptation techniques . . . 144
7.4 Comparison of Proposed with Existing Model(s) . . . 148
7.5 Confusion Matrix with Response Time for the Domain Adaptive Multimedia Event Detection Model using YOLOv3 . . . 150

8.1 Analysis of Related-Work with identified Requirements for Knowledge Transfer without Bounding Boxes . . . 158
8.2 The mean average precision (mAP) while using ILSVRC for Weak Level labels and Microsoft COCO & OID for Strong Level labels. . . . 169

Chapter 1

Introduction

1.1 Introduction

The Internet of Things (IoT) is designed to support intelligent systems for different domains to enhance the quality of life. Research in IoT is more focused on processing scalar data generated by various sensors within smart cities like smart energy events having readings from temperature sensors or energy sensors. Similarly, RFID tag readings from packet loss events are another example of scalar events, and IoT is more focused on such structured events. However, due to the increase in the shift of Internet traffic towards multimedia, sensors within smart cities also produce a huge amount of multimedia data [4] and thus require handling of a large number of subscriptions belonging to multiple domains of multimedia applications. It has been predicted internet video will represent 82% of all internet traffic, where internet video surveillance traffic will increase sevenfold. A report is shown in Fig. 1.1 that clearly illustrates the rise in multimedia traffic over global IP traffic. Thus there is a growing demand for efficient consumption of both scalar as well as multimedia events (i.e., images, video, and audio), which is also shifting the focus of research from conventional IoT to multimedia-based IoT (IoMT) [5–7]. As the concept of integrating multimedia with IoT is very recent, it is not standardized yet and needs to be investigated fundamentally along with the adaptation of domains.

Event processing systems [8] are introduced to serve as middleware between IoT and applications layer [9] and applicable only for scalar data events. On the other hand, real-time image based systems can provide high performance for multimedia events. They are still designed only for specific domains, have limited user expressibility and tiny (bounded) vocabulary. Advancements in Deep Neural Network (DNN) may support IoMT data but have the limitation of availability of trained classifiers for unseen concepts (subscriptions). The conventional trend is to train such high-performance models on

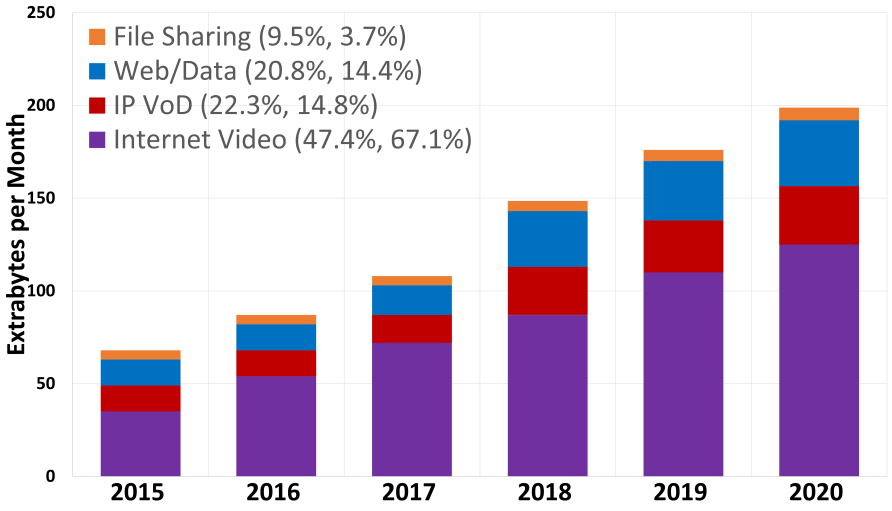


FIGURE 1.1: Global Internet Traffic by Application Category [1]

images with bounding box annotations for weeks and detect objects only for certain classes present in object detection datasets. Clearly, no settings are available for reducing the training time of object detection models on unseen categories. Since it is not practical to have all trained classifiers or training data available for a large number of classes, it is necessary to address the problem of training of classifiers online for unseen subscriptions with the provision of adaptation among domains, and ultimately minimizing the *response time* and increasing accuracy from the perspective of the user.

This thesis focuses on foundational aspects of the problem of an adaptive classifier based multimedia event processing which includes redefining *event processing* to *multimedia event processing*, defining *generalizability*, introducing image processing *operator* (like *detect* for object detection) for event query languages, standardization of the concept of *response-time*, identification of different *scenarios* of handling dynamic (seen/unseen) subscriptions, and established a fast online training detector for unseen concepts without bounding box annotations. I proposed a multimedia event processing model for the consumption of IoMT based data and optimized it at various levels moving from domain-specific to domain adaptation techniques. I developed prototypes for each of the identified scenarios of unseen subscriptions. The most specific work eventually builds up a detector that can train object detection models in 5-min while providing competitive accuracy without the need of bounding boxes for the training.

1.2 Problem Overview and Motivation

Event processing systems [8] are designed to process data streams and cannot natively include multimedia events produced by IoMT data. Event-based multimedia approaches

exhibit high performance in the current scenario but are designed for specific domains (like traffic management, security, supervision activities, terrorist attacks, natural hazards [10–14]) and hence can handle only familiar classes (have bounded/limited vocabulary). The escalating growth of multimedia data with large numbers of user subscriptions poses multiple challenges for the processing of IoMT based events. On the other hand, user subscriptions are also unseen (unknown) and may belong to various domains in smart cities' distributed environments. It is not evident in existing approaches to deal with such a large number of unseen concepts emerging and changing over time, typical for images/videos in the IoMT [15]. Furthermore, the essential requirement of multimedia applications is a real-time performance [16], which needs to be fulfilled for its usability. This highlights the need for minimization of *response time* while maintaining *accuracy* from the user's perspective. Presently, there is no provision of such generalized multimedia event processing that can handle seen/unseen subscriptions belonging to multiple domains to achieve high accuracy in low response-time.

If we have to support a generalizable approach to multimedia event processing with high performance, the event engine needs to support an extensive range of concepts/objects within subscriptions. Thus we realized the requirement of availability of trained classifiers to process multimedia events using neural network-based techniques in real-time. The online training of classifiers is an option, but the high cost of training [17, 18] limits the ability of the event engine to respond in a timely manner to new concepts. Current online learning-based approaches make their decisions on the fly [19–21]. Still, most of them are solely based on concept drift in multimedia streams and inapplicable for handling unseen subscriptions. Apart from the limitation of the availability of pre-trained classifiers, another weakness of traditional approaches is that even if we have trained classifiers available, we have to start from scratch when constructing another classifier. This is not satisfying and could be easier by applying the notion of *adaptation* among domains for optimization.

Optimization techniques in neural network models are based on the trade-off of speed and accuracy [22], which is supposed to be done before the processing of events and focuses only on accuracy and generalization ability of classifiers or on the computation cost, including testing time [23–25], excluding the training time of DNN models. Thus existing optimization techniques cannot be configured at run-time in case of adaptive subscriptions of multiple domains and need to be further investigated for minimizing the overall response time, including both testing and training time. Furthermore, the major challenge in training DNN based models (specifically for object detection) is the need to collect a large amount of images with bounding box annotations, which is not possible for thousands, or millions, of unseen classes. All of these shortcomings of multimedia event processing motivate us to address the problem of minimizing the response time of

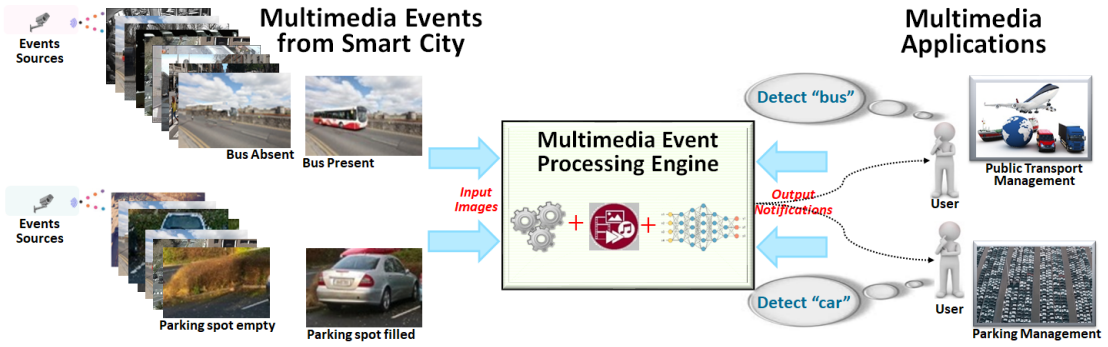


FIGURE 1.2: Generalizable Multimedia Event Processing

object detection models in responding to seen/unseen concepts-based queries, using an online domain adaptation based classifier construction approach while reducing the use of bounding boxes annotations and achieving high accuracy.

1.3 Motivational Scenario

Consider the scenario of object detection for analyzing multimedia events in smart cities (shown in Fig. 1.2). Suppose a user subscribes for the detection of "Bus" on "Bus Stand". This type of query can be answered "Public Transport Management" using a camera observing the bus stand and producing multimedia events consisting of bus status-related information. Similarly, if a user subscribes for the detection of the empty parking spot (i.e. absence of car at parking spot), we require another application for processing "Car Parking Management" events. Moreover, if a user subscribes to concepts like "taxi" or "pedestrian", then existing public transport and car parking management systems will not respond to any new class even if they already consist of similar classes like "car" and "person". Thus we need a generalizable multimedia event processing system that can provide adaptation from seen to unseen concepts (like *car* to *taxi*) and able to answer any completely unseen concept (like a *cat*, *dog*, *key*, *bicycle*, etc.) of any domain.

1.4 Problem Statement

This thesis focuses on answering user queries *online* consisting of seen (bounded vocabulary) as well as *unseen* concepts (unbounded vocabulary) that include processing of multimedia events while achieving high accuracy and minimizing the *response-time*, where the training of classifiers *may* or *may not* have *bounding box annotations* available? The concept is primarily based on the following four dimensions:

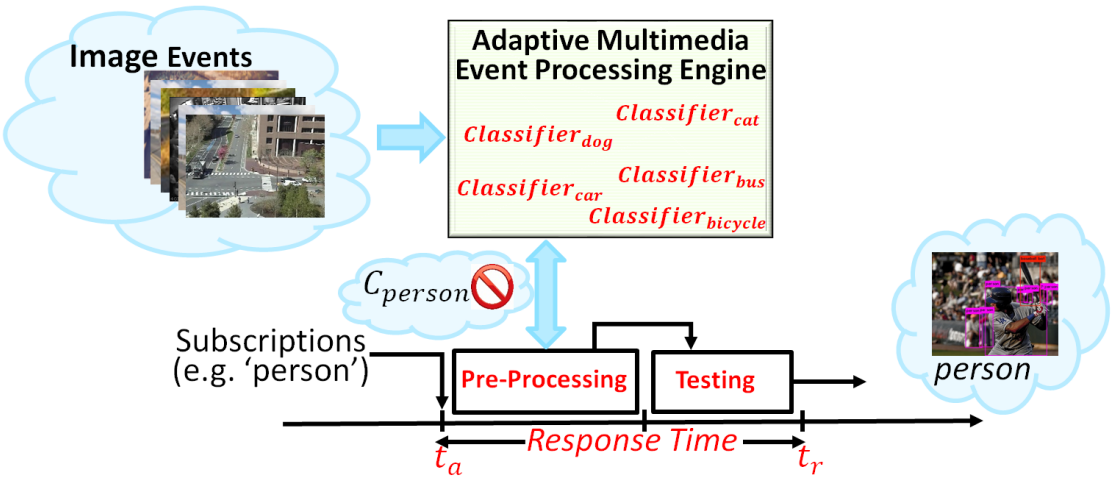


FIGURE 1.3: Response Time in Multimedia Event Processing

1. Support for Unbounded Vocabulary (Generalizability): This dimension concerns the ability to recognize a large number of seen/unseen subscriptions (presently keywords) for the naming of objects that may not belong to the bounded (limited) vocabulary of the system.

2. High Accuracy of Multimedia Processing Method: It deals with the high *accuracy* of the feature extraction method (presently object detection) within allowable response-time, used to support unstructured multimedia events that may contain information in the form of images or video.

3. Low Response-Time: The first dimension is related to the objective function "response-time" that needs to be minimized for the standardization of the problem. Response time (as shown in Fig. 1.3) is defined as the time difference between the time (t_a) the subscription arrived and the time (t_r) at which the system is ready to notify the subscriber. We define the pre-processing stage of the multimedia event processing system by considering mainly two cases:

   - Case 1: Classifier for Subscription Available (Seen Concept)
   - Case 2: Classifier for Subscription Not Available (Unseen Concept)
     - Case 2(a): Subscriptions require classifiers similar to base classifiers
     - Case 2(b): Subscriptions require classifiers completely different from base classifiers

   Based on the availability of the type of training data for t_{dc}, we can further classify the present case of unseen concepts into the following two scenarios:

   - Object-Level Annotations Available: This scenario assumes we can collect images with bounding box annotations using existing object detection datasets

[26–28]. However, all of these datasets have bounded vocabulary having a finite number of classes. Thus it is not possible to provide bounding box labels for thousands, or millions, of classes. Moreover, it is much easier to offer image-level annotations.

- Image-Level Annotations Available: In this case, we assume that only image labels are available with no bounding boxes. Such image-level annotations can be obtained using online data collection toolkits[1] or image tags on any (i.e., Flickr, Google, and/or Bing) image web search.

4. Support for Domain Adaptation (Maintainability): It refers to the ease of transfer to multiple domains with less manual effort. We treat the transformation of any classifiers into detectors as a domain adaptation task, i.e., transfer from source (full image recognition) to target (localized recognition) domain as a domain adaptation problem. Besides the adaptation of classifiers into detectors, the system should support adaptation between visual domains (like bus→car, dog→cat).

1.5 Proposed Approach

The problem of multimedia event processing is divided into different scenarios shown in Fig. 1.4. When a user subscribes to any new concept, the multimedia event processing engine should recognize whether the model is previously seen (familiar) or do we need to construct a new classifier for it. Scenario 0 handles the seen concept using the proposed framework of multimedia event processing and baseline classifiers trained offline available in Model-I "Domain-Specific Classifier based Multimedia Event Detection". However, if subscription consists of an unseen concept, then the model tries to find any similar concept available for the knowledge transfer. In the case of a completely unseen concept (Scenario 1), there is a need to train classifiers from scratch with optimization techniques discussed in the Model II "Hyper-Parameters based Adaptive Multimedia Event Detection".

If the unseen concept has similarities with the seen concept, then the last condition checks the accessibility of bounding boxes because existing bounding box annotations-based datasets consist of a limited number of concepts. It is essential to note that the comprehensive similarity scores between seen and unseen concepts are computed using semantic and visual similarities. Scenario 2 and 3 handle the partial unseen concepts where difference is the presence or absence of bounding box annotations for the training of classifiers. The proposed approach for processing unseen concepts from existing seen

[1] https://github.com/tzutalin/ImageNet_Utils

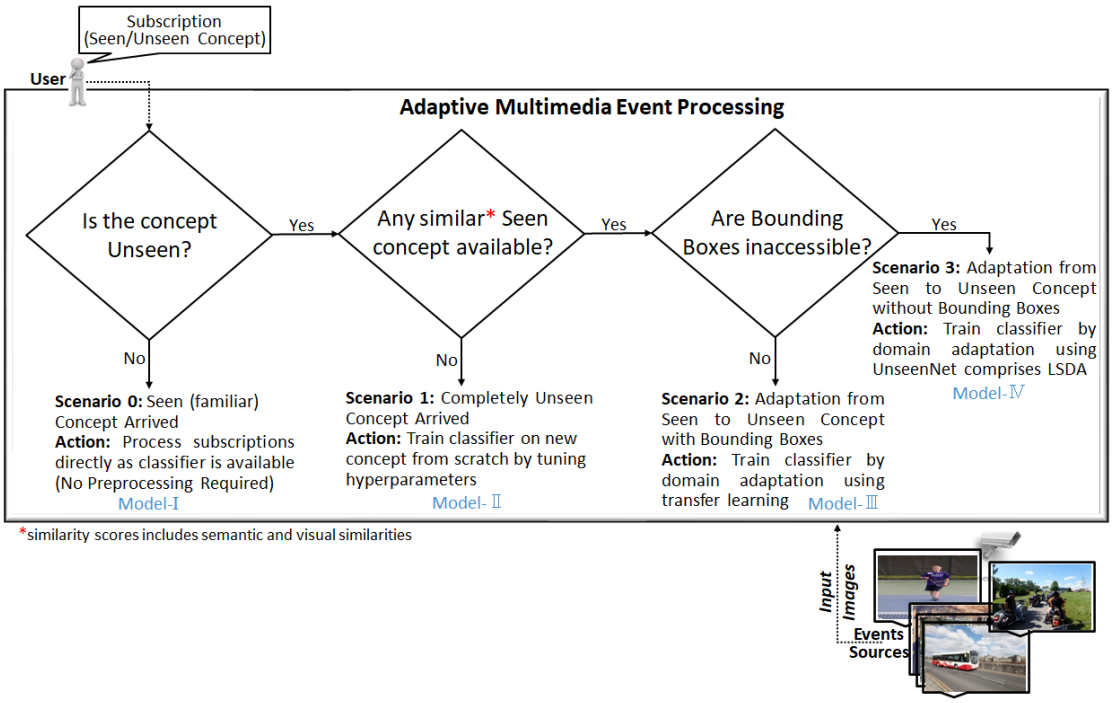


FIGURE 1.4: Scenarios for Multimedia Event Processing adhering to Seen/Unseen Concept Problem

concepts with the transfer of knowledge is presented in Model III, "Domain Adaptation based Multimedia Event Detection". Along with transfer learning, the proposed Model IV "Domain Adaptation-based Multimedia Event Detection without Bounding Boxes" utilizes weakly supervised learning to eliminate the need for object-level labels and classifiers could be trained without bounding boxes.

Table 1.1 summarizes the proposed work for covering the sub-problems of multimedia event processing. I introduce multimedia as a native event type in event processing and uses deep neural network-based models to optimize the framework (applicable for Scenario 0 presented in Model I). Object detection is used in this thesis for analyzing multimedia events to demonstrate the efficiency and limitations of proposed models. Next, the proposed model can handle completely unseen concepts by training from scratch with adaptive hyperparameter tuning-based adaptation (Scenario 1, Model II). An enhancement in performance is presented by introducing domain adaptation for the unseen concepts having similarities with seen concepts (Scenario 2, Model III). In the last specific problem, I finally removed the limitation of bounding box annotations to train unseen concepts (Scenario 3, Model IV). Multiple approaches of proposed models with their specific contributions are given below in this section.

TABLE 1.1: Summarizing proposed work with analyzed sub problems of multimedia event processing.

Problem Coverage	**Scenario**	**Approach**	**Model**
Introduce multimedia as a native event type in event processing with high performance.	Seen Concept (Scenario 0)	Event Processing + Multimedia Analysis + Deep Neural Networks	Domain-Specific Classifier based Multimedia Event Detection (Model I)
Support of completely unseen concepts online with high performance.	Completely Unseen Concept (Scenario 1)	Event Processing + Object Detection+ Adaptation (Hyperparameter Tuning)	Hyper-Parameters based Adaptive Multimedia Event Detection (Model II)
Enhance the performance for the unseen concepts that have similarities with seen concepts.	Partially Unseen Concept (Scenario 2)	Event Processing + Object Detection + Domain Adaptation (Transfer Learning)	Domain Adaptation based Multimedia Event Detection (Model III)
Enhance the performance for the unseen concepts that have similarities with seen concepts while eliminating the requirement of bounding box annotations.	Partially Unseen Concept and Bounding Boxes Inaccessible (Scenario 3)	Event Processing+ Object Detection + Domain Adaptation (Transfer Learning + Weakly Supervised Learning)	Domain Adaptation based Multimedia Event Detection without Bounding Boxes (Model IV)

1.5.1 Domain-Specific Classifier based Multimedia Event Detection

The first model is designed to handle problem dimensions of *generalizability*, high *accuracy*, and low execution time (part of *response-time*). The proposed system first incorporates the event-based system with multimedia analysis to support the processing of IoMT based event streams within the publish-subscribe paradigm and allow the inclusion of new operators using deep convolutional neural network (DNN) based techniques. A new "detect" operator has been developed to provide the requisite DNN based feature extraction to detect objects inside image events. Conventionally DNN based methods are dependent on the trained classifiers. These classifiers are trained on general-purpose datasets consisting of a large number of classes, which may reduce the performance. Thus, we also proposed a subscription-based optimization technique, "Classifier Division and Selection," which relies on the division of classifiers based on domain and selection of classifiers based on subscriptions. Experiments show that the

proposed system can achieve an average throughput of 110 frames/sec with an approximate accuracy of 66.34% with a permissible response-time of 2 hours of training on real-world events. Moreover, the ability to process multiple domains signifies its generalizability on various applications of smart cities. Experiments of all models (I to IV) have been conducted on Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0-26-generic x86_64), with NVIDIA TITAN Xp GPU.

1.5.2 Hyper-Parameters based Adaptive Multimedia Event Detection

In the previous model, we realized that DNN based techniques are effective for processing image events, and the limitation of having to train classifiers for unseen concepts may increase the overall response time. Thus to tackle the dimension of supporting *unseen* subscriptions with minimized *response-time*, we proposed an online classifier construction based model that can adapt among dynamic subscriptions with low response time and provide reasonable accuracy for the multimedia event processing. We optimized the multimedia event processing model by leveraging the hyperparameter tuning based technique, which analyzes the accuracy-time trade-off of object detection models and configures *learning-rate*, *batch-size*, and the *number of epochs*, using response-time based strategies: *Minimum Response Time needed while Minimum Accuracy allowed*, *Optimal Response Time needed while Optimal Accuracy allowed*, and *Maximum Response Time allowed while Maximum Accuracy needed*, for the dynamic subscription constraints. Our results indicate that the proposed online classifier training-based model can achieve an accuracy of 79.00% with 15-min training and 84.28% with 1-hour training even from scratch to process multimedia events.

1.5.3 Domain Adaptation based Multimedia Event Detection

Further, it can be argued that the concepts (presently classes) in real-world events are related to each other, which necessitates the notion of adaptation among domains for *generalizability* instead of training each classifier from scratch. Moreover, easy transfers among domains can also enhance *accuracy* while maintaining or minimizing the *response-time* dimensions. This model extends an adaptive multimedia event processing model by leveraging transfer learning-based techniques to provide domain adaptation among unseen concepts. We have also instantiated the online classifier learning model by transferring knowledge among classifiers using fine-tuning and freezing layers of neural network-based object detection models. Our investigation shows that the online training of object detection methods with transfer learning is helpful for accurate multimedia event processing and can provide faster results with a reduced response time for

the related domain-based subscriptions. The proposed domain adaptation-based online learning model can achieve an accuracy of 95.14% within 30-min of training. This work also confirms our model's suitability for any number of unseen concepts in subscriptions; however, this scenario assumes we can collect images for training with bounding box annotations.

1.5.4 Domain Adaptation based Multimedia Event Detection without Bounding Boxes

In the previous model, we assume we have object-level annotations (i.e., annotated bounding boxes) available for training the network. Since image-level annotations (i.e., without bounding boxes) are comparatively easy to acquire, we intend to improve our model further using our approach, "UnseenNet", in which we will not require bounding boxes for the training of classifiers and expect a decrease in response time. In this model, first, we train two baseline detectors (Strong Baseline and Weak Baseline) offline using existing object detection datasets (like Pascal VOC, OID, Microsoft COCO) and image classification dataset (like ImageNet), respectively. On request of any *unseen* concept, first, "UnseenNet" download images from the web (like Google/Bing Images) using only image-level labels (like goat). Strong Baseline Detector is then fine-tuned on collected images of unseen concepts by labeling the most semantically similar class (like sheep) with the unseen class name (like goat). At this stage, we also compute the visual similarity of the constructed unseen class detector (trained on classification data) with seen classes of *weak baseline detector*, combine it with semantic similarities, and select top-k classes ranked on comprehensive similarities. Here, semantic similarity refers to the similarity between the labels of classes using WordNet [29] and visual similarity of the Euclidean distance between weights of different classes. Finally, we transfer the knowledge of classifier-detector differences of top classes to the constructed unseen class detector and adapt it into the stronger detector for an unseen class without further training. Indeed, the idea of transformation of classifiers into detectors is utilized from knowledge transfer based LSDA methods [3, 30]. Our approach "UnseenNet" also makes use of MobileNetv2 in place of AlexNet for classification. It also takes advantage of much faster object detection models like YOLOv3 (compared to RCNN in the LSDA). Our model achieves a mean average precision (mAP) of 19.82 within 5-min of training, where existing frameworks could take >5.5 hours.

1.6 Research Hypothesis

The presented research is based on following main hypotheses:

1. **Research Hypothesis I:** Domain-Specific classifier based multimedia event processing assumes that if we construct N-Class classifiers for different domains, and we use subscription constraints to choose closely related classifiers for the processing of multimedia events; the performance will be enhanced in terms of *accuracy* and *response time*, and will also add the ability to generalize for multiple domains.

2. **Research Hypothesis II:** Hyper-Parameters based Adaptive Multimedia Event Detection model interprets the hypothesis "if tuning of hyperparameters based technique is useful in machine learning to improve performance; then performance will also get enhanced for low *response-time* even on training from scratch for *unseen* subscriptions on tuning hyperparameters for the online construction of classifiers." Here, performance indicates speed-up in training and increment in accuracy.

3. **Research Hypothesis III:** Domain adaptation based Multimedia Event Detection model relies on the fact that if transferring of knowledge from one domain to another (say $A \rightarrow B$) can improve the performance as compared to fine-tuning of pre-trained models (like $C_{P_{ImageNet \rightarrow B}}$) or training of classifier from scratch (C_B); then there will always be a decrease in *response-time* with increase in the *accuracy* of constructed classifier ($C_{A \rightarrow B}$) compare to the classifier trained from pretrained model (like $C_{P_{ImageNet \rightarrow B}}$) or training from scratch (C_B).

4. **Research Hypothesis IV:** The approach of "Domain Adaptation based Multimedia Event Detection without Bounding Boxes" based on the hypothesis "if an adaptation of classifier into detector eliminates the need of bounding boxes as well as transferring of knowledge from one domain to another speed-up the training; and a detector gets constructed from classifier with the help of transfer of knowledge from visually/semantically similar classifier; then that detector will take less time to train for unseen classes and eliminate the requirement of bounding boxes".

1.7 Research Methodology

The research methodology followed in this work consist of the following main steps:

1. Comprehensive literature review of four main areas: event processing, multimedia analysis, domain adaptation, and object detection.

2. Formulation of the problem of processing multimedia events for dynamic (seen/unseen) concepts.

3. Identification of core requirements for the problem of generalized multimedia event processing.
4. Introducing the concept of response-time for standardization in event-based models to support multimedia events consisting of seen/unseen concepts.
5. Breaking down of problem in specific research questions for the establishment of generalized multimedia event processing.
6. State the hypothesis to answer each research question.
7. Designing of experiments while analyzing required neural-network based models and object detection datasets.
8. Investigation of current approaches specific to particular research questions for the purpose of comparison.
9. Implementation of proposed models.
10. Analysis of the optimized results and conclusions.
11. Repeating the same steps $6 - 10$ for all (presently four) hypotheses.
12. Reporting of limitations of the proposed approach with possible future directions.

1.8 Core Contributions

The contributions of this research are manifold:

- Analyzing trends, challenges, and opportunities for the generalized multimedia event processing based applications using IoMT by considering object detection as a case study [31].
- Formulation of the problem of processing multimedia events for dynamic subscriptions using domain-specific classifiers, online training, and transfer learning based large scale domain adaptation approaches, for covering the requirement of *generalizability* and supporting *seen/unseen* subscriptions [32–34].
- A neural network based event matcher optimized using subscription constraints for the feature extraction, with the provision of "detect" operator in event query languages to support object detection in multimedia events, is proposed in the *domain-specific classifier based multimedia event detection* model to increase the *accuracy* and low *response-time* [32].

- Standardization of objective function "Response-Time" for the *domain adaptation based multimedia event detection* and providing response-time based strategies with their respective prototypes by tuning *hyperparameters* for the real-time classifier training [34].
- An adaptive architecture for online classifier construction with the aim of minimizing the *response-time* and maximizing the *accuracy* also proposed with *hyperparameters based multimedia event detection* model [33].
- An instantiation of the online classifier learning model by transferring knowledge among classifiers using fine-tuning and freezing layers of neural network-based object detection models is also shown in the *domain adaptation based multimedia event detection* model [34].
- Enhancement for the performance of object detection models (YOLO, SSD, and RetinaNet [35–37]) on multimedia events and seen concepts belonging to Pascal VOC, Microsoft COCO, and OpenImages datasets [26–28], which achieves
  - accuracy of 66.34% with permissible response-time of 2-hours for unseen subscriptions while using subscription based classifier selection approach in *domain-specific classifier based multimedia event detection* [32].
  - accuracy of 84.28% within 1-hour response-time for unseen subscriptions, by using online classifier construction from scratch based approach using *hyperparameter tuning* [33].
  - accuracy of 95.14% within 30-min of response-time for unseen subscriptions while using *online domain adaptation of classifiers based approach* [34].
- UnseenNet, a LSDA based detector for the training of unseen classes using only image-level labels with no bounding boxes annotations by using the fastest classification and detection models while utilizing object detection and image classification datasets having a limited vocabulary [38].
- While devising a fast detector *UnseenNet*, we also derive the limits of *response-time* from 5-min to 20-min (on GPU) in the area of weakly supervised learning (i.e., training with no bounding boxes), where existing frameworks take >5.5-hours to attain similar mAP.

1.9 Thesis Organization

The remainder of the thesis is organized as follows:

- Chapter 2 – Problem Formulation: This chapter first presents the background knowledge required to formulate the problem. Second, it highlights the limitations and challenges to motivate the problem. It also covers problem formulation for response-time and research questions of this work. Some of this work has been published in [31, 32].

- Chapter 3 – Background and Related Work: This chapter provides a detailed discussion of the Internet of Multimedia Things with its state of the art. Moreover, it provides background and comparison of existing deep neural network-based object detection models. Some of this work has been published in [31].

- Chapter 4 – Adaptive Multimedia Event Processing: This chapter states the scenarios of handling dynamic subscriptions of multimedia event processing and provides a brief overview of four main models designed to interpret different adaptive multimedia event processing scenarios.

- Chapter 5 – Domain-Specific Classifier based Multimedia Event Detection: This chapter focuses on extending event processing languages with the introduction of operators for multimedia analysis and leverages subscription constraints in-order to optimize the deep convolutional neural network-based event matcher for research Hypothesis-I. It details the proposed model, implementation algorithms, evaluations, and results. This work has been published in [32].

- Chapter 6 – Hyper-Parameters based Adaptive Multimedia Event Detection: This chapter investigates an online classifier construction approach that can handle unseen dynamic concepts with high performance. Also, demonstrate that the deep neural network-based object detection models with hyperparameter tuning can improve the accuracy within less training time. Finally, it details the proposed model for the research Hypothesis-II, implementation algorithms, evaluations, and results. This work has been published in [33].

- Chapter 7 – Domain Adaptation based Multimedia Event Detection: This chapter focused on research Hypothesis-III and proposed an online training approach of deep neural network-based object detection methods with transfer learning to reduce further the response time and increase accuracy for unseen concepts. Lastly, it details the implementation algorithms with experiments and results. This work has been published in [34].

- Chapter 8 – Domain Adaptation based Multimedia Event Detection without Bounding Boxes: This chapter continues the investigation of the third research Hypothesis-IV and provides details of the proposed UnseenNet (LSDA based model) for processing unseen concepts without annotated bounding boxes, implementation algorithms, evaluations, and results. Some of this work has been published in [34] and currently under submission.

- Chapter 9 – Conclusion and Future Work: Finally, this chapter highlights the concluding remarks and the limitations of the proposed multimedia event processing based adaptation models and avenues for future work.

1.10 Summary of Conclusions

An adaptive approach for multimedia event processing has been proposed in this work, using domain knowledge transfers while online classifier construction of object detection models to handle unseen concepts (with/without bounding boxes) in low response-time. The proposed model has been optimized at various stages using *classifier division and selection*, *tuning of hyperparameters*, and *transfer of domains* based techniques. The performance is enhanced from 0 to95.15% in terms of accuracy for 30-min of response time to train unseen concepts. Finally, we proposed an "UnseenNet" detector for training object detection models without bounding boxes that achieves a mean average precision (mAP) of 19.82 within 5-min of training, where existing frameworks take >5.5 hours.

1.11 Associated Publications

The below list represents the different aspects of the research being published/in-review during the course of this thesis:

- Asra Aslam, Souleiman Hasan, and Edward Curry, "Challenges with image event processing: Poster," In Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems, pp. 347-348, ACM, 2017. (**Conference Rank: B**)

- Asra Aslam and Edward Curry, "Towards a generalized approach for deep neural network based event processing for the internet of multimedia things," IEEE Access 6 (2018): 25573-25587. (**Journal Impact Factor: 4.098**)

- Asra Aslam and Edward Curry, "A Survey on Object Detection for the Internet of Multimedia Things (IoMT) using Deep Learning and Event-based Middleware:

Approaches, Challenges, and Future Directions", Image and Vision Computing (IMAVIS), Elsevier (**Journal Impact Factor: 3.012**)

- Asra Aslam and Edward Curry, "Investigating Response Time and Accuracy in Online Classifier Learning for the Multimedia based Publish-Subscribe", Multimedia Tools and Applications (MTAP), Springer (**Journal Impact Factor: 2.313**)
- Asra Aslam and Edward Curry, "Reducing Response Time for Multimedia Event Processing using Domain Adaptation." Proceedings of the 2020 International Conference on Multimedia Retrieval. (**Conference Rank: A2**)
- Asra Aslam and Edward Curry, "Detecting Seen/Unseen Concepts while Reducing Response Time using Domain Transfer in Multimedia Event Processing", ***Submitting*** to IEEE Access (**Journal Impact Factor: 4.098**)
- Asra Aslam, "Object Detection for Unseen Domains while Reducing Response Time using Knowledge Transfer in Multimedia Event Processing." Proceedings of the 2020 International Conference on Multimedia Retrieval (**Conference Rank: A2**)
- Asra Aslam and Edward Curry, "UnseenNet: LSDA-based Fast Training Detector for Unseen Concepts with No Bounding Boxes", ***Submitting*** to IEEE Transactions on Pattern Analysis and Machine Intelligence (**Journal Impact Factor: 17.861**)

Chapter 2

Problem Formulation

2.1 Introduction

This chapter provides an overview of the foundational aspects of the problem addressed in the thesis. The enormous generation of multimedia data within smart cities belonging to an increasing number of applications imposes a requirement of efficient handling of multimedia-based events. This multimedia data could require processing millions of seen/unseen concepts (like person, cat, dog, car, bus, kid, etc.), which states the need for generalizable multimedia event processing specifically to detect objects in smart cities. A brief discussion on the problem domain with technical limitations is presented in Section–2.2.

The drawbacks of existing IoT and need of IoMT based event processing is presented in motivation Section–2.3. I formulate requirements for the comparison of related work in Section–2.4. Moreover, I summarize the challenges and future research directions for the generalizable multimedia event processing (by taking object detection as an example) in Section–2.5. Finally, I divide the problem statement "How can we answer user queries *online* consisting of seen (bounded vocabulary) as well as *unseen* subscriptions (unbounded vocabulary) that include processing of multimedia events while achieving high accuracy and minimizing the *response-time*, where the training of classifiers *may* or *may not have bounding box annotations* available?" in specific research questions in Section–2.7) and define response-time formally in Section–2.6.

2.2 Problem Domain and Technical Limitations

Recently, leveraging the Internet of Things (IoT) to process information related to various large-scale real-time data processing applications is becoming a popular trend in the proliferation of smart cities [39]. IoT infrastructures are well established, consist of adequate communication, efficient processing protocols, and optimization techniques. IoT middleware is responsible for providing shared services to applications and eases the development process. In the current scenario, research in IoT mainly focuses on handling the challenges of big data, excluding multimedia, leaving a gap between the advancement of IoT and multimedia-based technologies. However, the IoT cannot realize the goal of interconnected objects unless it includes "multimedia" within the processing of information to analyze the Internet of Multimedia Things (IoMT) based events.

Event processing systems are designed to process the subscription of a user based on standard languages in response to events. Popular event-based approaches rely on a publish-subscribe paradigm while utilizing a mediator for providing services and works for supporting application-specific structures. It is observed that existing publish-subscribe based event processing systems only focus on structured (scalar) events for the processing of subscriptions of a user, with no provision of handling multimedia data. The high-speed nature of event streams with high bandwidth of multimedia data also requires optimization strategies. However, optimization techniques in event processing systems are generally based on predicate indexing and network algorithms of matching subscriptions.

Numerous applications are designed for processing multimedia (unstructured) systems events with high efficiency and applicable only for specific roles. For instance, traffic control, health monitoring, parking management, or any surveillance applications shown in Fig. 2.1. Such high-performance applications process only their specific (familiar) "seen" concepts and cannot process any new/unseen concept. Moreover, they possess variance in performance, moving from one application to another. We assume high-performance deep neural network-based models could be a possible solution for generalizable multimedia event processing. However, it is not practical to construct classifiers for unbounded vocabulary consisting of millions of categories like person, cat, lion, horse, bike, car, taxi, bus, etc. Thus, deep learning methods impose the constraint of ***training*** of classifiers for "unseen" classes before the matching of multimedia events. Moreover, there is no consideration for the duration of training time of classifiers to reduce the overall response time. Response time is defined as the difference between the arrival and notification time of subscription (detailed in Section–2.6). We aim to minimize this response-time (Fig. 2.1) for "unseen" concepts while achieving high accuracy.

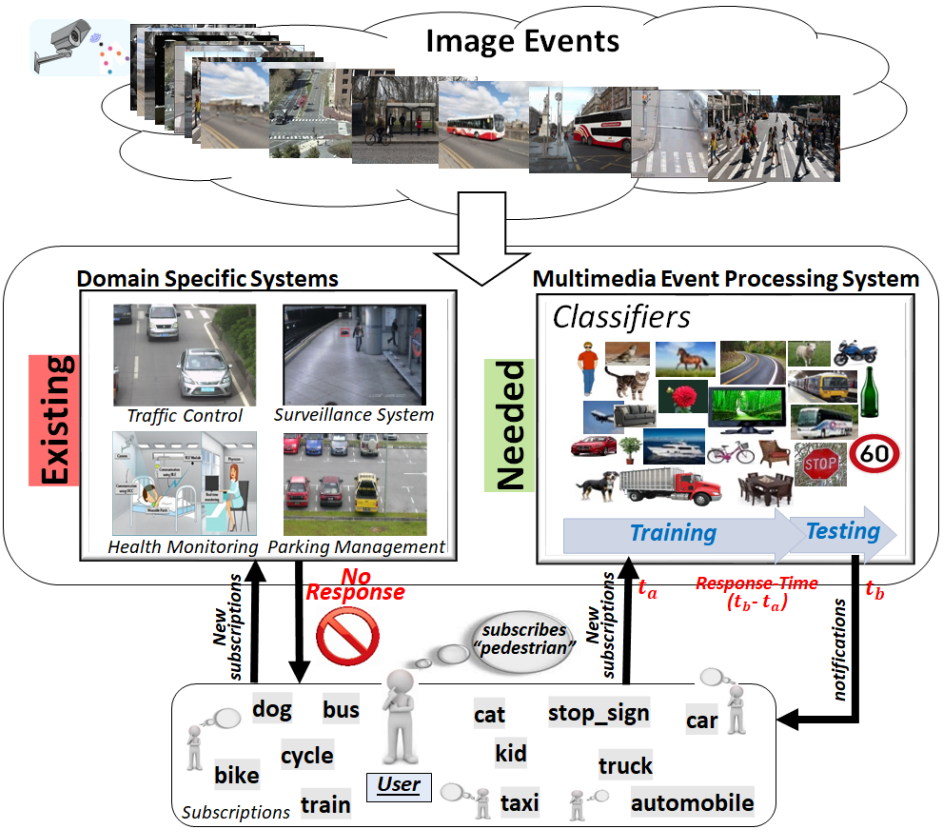


FIGURE 2.1: Problem analysis of Multimedia Event Processing using Online Training of Classifiers for Seen/Unseen Objects

Furthermore, along with classifiers' training, we also realized the need for the availability of training-data for unseen categories. In the present scenario, we have object detection datasets available for training. Since the collection of bounding boxes with images is tedious, they have a small number of classes or fewer images per class. Moreover, since image-level labels are comparatively easy to acquire, a large number of classes can be covered easily using image classification datasets or from the web. However, image classification datasets have only image-level labels (i.e., images with only labels) and no bounding box annotations. In this work, our objective is to make use of these small number of classes based object detection datasets (having bounding boxes annotations) and a large number of classes based image classification datasets (have no bounding boxes annotations) and convert them into infinite (unbounded) vocabulary based classifiers while limiting the training time.

Finally, we state our problem as "How can we answer user queries *online* consisting of seen (bounded vocabulary) as well as *unseen* subscriptions (unbounded vocabulary) that include processing of multimedia events while achieving high accuracy and minimizing the *response-time*, where the training of classifiers *may* or *may not have bounding box annotations* available?"

2.3 Motivation

Multimedia communication is gradually becoming an essential source of information in multiple scenarios, including traffic management, security, supervision activities, terrorist attacks, and natural hazards. This enormous generation of multimedia data within smart environments with an increasing number of applications requires efficient handling of multimedia-based events. Moreover, users' subscriptions may vary from one domain to another and require the processing of millions of such dynamic seen/unseen concepts. Furthermore, the essential requirement of multimedia applications is real-time performance [16], which needs to be fulfilled for its usability. This highlights the need for minimization of response time while maintaining accuracy from the perspective of the user. These drawbacks form the underlying motivation for the presented work, where the proposed online classifier training-based multimedia event processing engine utilizes the publish-subscribe paradigm and leverages neural network-based object detection methods to meet the requirements of dynamic subscriptions.

2.3.1 Scenarios

Since image recognition is the most common challenge in the context of smart cities, consider scenarios of object detection (shown in Fig. 2.2) for analyzing real-time multimedia events. Users can further utilize the information associated with public transport and parking-related events with the following subscribed query statements:

Example 1: Public Transport Management

Suppose a user subscribes for the detection of "Bus" on *Bus Stand?* and we have a camera observing the bus stand. The multimedia events produced by such sensors (like the camera) consist of bus status-related information (see Fig. 2.2(a)), as the bus has already arrived at the bus stand in the first image, and there is no bus in the second image. Currently, processing such bus related queries will need a domain specific application like public transport management. Moreover, any change in the query (say Is "Taxi" on *Taxi Stand*?) may require a different domain specific application. Thus, we need a generalizable system that can automatically adapt from one domain to another without manual effort of generating domain-specific applications. In that case, it will be easy for users to monitor such situations through multimedia event-based queries without learning a limited vocabulary of specific application domains.

Example 2: Car Parking

Consider the case of car parking events; if a user subscribes for the detection of


(a) (i) Bus present; and (ii) Bus absent in image event


(b) (i) Car present; and (ii) Car absent in image event

FIGURE 2.2: Multimedia Processing Events Scenario

parking spot empty i.e., absence of car at the parking spot, this type of query can also be answered using multimedia events (shown in Fig. 2.2(b)) related to the parking status of one parking lot with the presence and absence of the car. In the current scenario, such queries may require a specialized "parking management system". However, a generalizable system can handle such new/unseen concepts because of its unbounded vocabulary and no requirement of construction of new domain-specific systems.

2.3.2 Generalizability

I assume introducing generalizability in IoMT based systems is crucial for the processing of multimedia data. Generalizability is the term used to provide flexibility in processing type of data: structured or multimedia. Generalizability also focuses on the ability to handle all possible domains by transferring knowledge from one domain to another. This is also referred to as *domain generalization* [40]. Moreover, the functionalities of providing different types of operations on multimedia data other than only object detection can also make the proposed system generalizable. I define generalizability formally with examples specifically for the proposed multimedia event processing in Section–4.2.1.

2.4 Requirements

In order to provide the high-level guidance for the construction of real-time stream processing based applications, eight core benchmark requirements have been suggested in the paper [41]. These requirements are based on effective and efficient processing of events, like *Keep the Data Moving*, *Integrated Stored and Streaming Data*, and *Process and Respond Instantaneously* are highly focused on achieving low latency. Processing of real-time streams without the need of costly storage operation is the key of "Rule: Keep the Data Moving" which requires an active event processing model and ultimately low inference time for the case of processing multimedia events. Similarly the "Rule: Integrated Stored and Streaming Data" also relies on efficiently storing, accessing, and modifying state information, which moderately directs the proposed multimedia event processing engine towards adaptation of available classifiers, and low domain adaptation time eventually. Finally, highly-optimized processing with minimal-overhead execution is the crucial requirement for any event processing system, realized in "Rule: Process and Respond Instantaneously" and should also be supported by multimedia stream processing. Thus these demands of achieving low inference, adaptation, and overhead time in real-time stream processing, lead us to define the following requirements to process multimedia events.

- Low System *Response-Time*, defined as the time difference between subscription arrived and time at which the system is ready to respond with the required accuracy. The requirement of low response time is needed to simulate real-time applications; thus, we aim to provide a fast response that will be independent of user subscriptions domains.
- High *Accuracy* of feature extraction method (presently object detection) on multimedia events within an allowable response time.
- Support for *Large Vocabulary* is the ability to recognize a large number of keywords for the naming of objects that may not belong to the limited vocabulary of the system.
- *Maintainability* refers to the ease of transfer to multiple domains with less manual effort.

2.5 Challenges and Opportunities

2.5.1 Challenges

- Standardization of the concept IoMT: We analyzed that existing IoMT based applications are domain-specific, and standardized IoMT architecture needs to be investigated. Seng et al. [4] and Almajali et al. [6] initiated the standardization of architecture for Multimedia Internet of Things while handling the issues of multimodal Big data computation, with scalability and maintainability of the model for effective multimedia information sharing [6]. However, it is an emerging challenge that needs more attention from different IoT and multimedia communities to agree on and suppose to cover all of the requirements of IoMT based systems of smart cities.

- Generalizability: As existing methods of multimedia event processing are domain-specific, generalizability (defined in Section–2.3) is another challenge for IoMT based systems. It is also recognized as domain generalization in literature [40, 42, 43]. For example, a system that can recognize a *bus* in "Public Transport Management" cannot recognize a *car* for the "Parking Management System". Similarly, the parking system cannot recognize "taxi" or "pedestrian". Thus, the task of moving from one domain to another is challenging in smart cities. Each of these systems re-implement middlewares, user interface, multimedia processing methods (like object detection models), etc. Furthermore, such systems also need to be integrated with IoT for deployment in smart cities. Therefore, the construction of a generalizable system for all applications that do not require such re-implementations each time with a domain change is an open question and one of the significant challenges of IoMT based systems.

- More Training Time: Presently, all of the current object detection models are compared only based on inference time and accuracy [44–47]. However, to deploy them in smart cities, we need to train on new classes/ scenarios in real-time. There is no research or comparison based on the training time of these neural network-based models to the best of our knowledge. This will result in the first response time of these object detection models will always be very high (maybe in days). Therefore, we believe there is a great need for such models to reduce training time rather than focus only on accuracy.

- Training Data Availability: We also conclude that even object detection models are accurate and fast. They are data-hungry, and they will never be able to perform better without sufficient data for different applications of smart cities. We need to understand that the maximum number of classes in specifically object

detection datasets is only 600 [28], and we see millions of applications for detecting multimedia events in smart cities. This is somehow not-satisfiable and unintuitive that no single dataset successfully provides sufficient classes for the detection of objects in smart cities. Nevertheless, we consider this as the biggest challenging problem and practically impossible from the current approach of construction of datasets in terms of cost, memory, time, resources, etc.

2.5.2 Opportunities

- Multimedia aware Middlewares: Event-based middlewares are a well-known solution for IoT that abstracts the complexities of the system/hardware from the application developer [48, 49]. Such existing event-based middlewares consist of rich literature for structured event processing and managed to bring an uprising change in the communication models of distributed systems. Thus we can firmly assume the success of multimedia events-based middlewares for the multimedia-based IoT approaches.

- Deep Neural Network (DNN) based Models: Deep learning has made significant progress in image recognition and opens a new path for the surveillance applications of smart cities [50–52]. The inclusion of deep convolutional networks based techniques for multimedia analysis of events could be a possible future solution for the standardization of IoMT methodology. A generalized approach for the IoMT data is realized in work [32] and demonstrates the proficiency of deep neural networks in processing multimedia event streams of multiple applications. The ability of DNNs to deliver high performance and continuous learning ability can effectively improve the demands of multimedia in IoT [53–55]. Irrespective of providing high-performance capabilities in image recognition, DNN-based techniques may include any classifiers to facilitate different kinds of applications in smart cities.

- Online Domain Adaptation: We observed it is not evident in existing approaches on how to handle a large number of concepts emerging from different applications of smart cities. Also, it is not possible to construct all classifiers from all specific domains consisting of all specific classes of smart cities. However, the field of domain adaptation is showing its benefits in detecting objects with real-time performance using knowledge transfers [56, 57]. We expect that IoMT based systems should incorporate these domain adaptation methods to construct classifiers for new/unseen concepts while applying transfer learning on seen concepts [15, 58–60]. Classifiers for seen concepts could be built from existing object detection datasets and serve as base classifiers for unseen concepts. Moreover, we have plenty of ontologies available for conceptual mapping relationships among classes.

For instance, WordNet [29] is used for semantic relationships, or we could also use existing visual relationship-based methods that appear in recent works [30, 61].

- Online Training: Besides online domain adaptation, we may need to train classifiers for entirely unseen concepts for any application of smart cities [34, 62, 63]. We can use online data collection[1] and automatic annotation[2] techniques to construct such a classifier. We may need to use automated data collection or annotation techniques in a worse scenario. Presently online data collection toolkits can download images for training using concept names; however, these images are iconic and consist of no bounding boxes. In such cases, we could improve accuracy for IoMT based data using semi-supervised or unsupervised models in different applications of smart cities [55, 64, 65].

- Training without Bounding Boxes: Suppose we can only collect training data that does not have bounding boxes to train classifiers of new applications. In that case, fortunately, we can apply new LSDA (Large Scale Detection through Adaptation) based methods that do not require bounding boxes for effective training [3, 30, 38, 66]. These LSDA based methods are designed to construct classifiers on new concepts for which we do not have sufficient data or no-annotated data. We believe bringing such baseline methods online could improve the limitations of processing IoMT based data using deep neural networks.

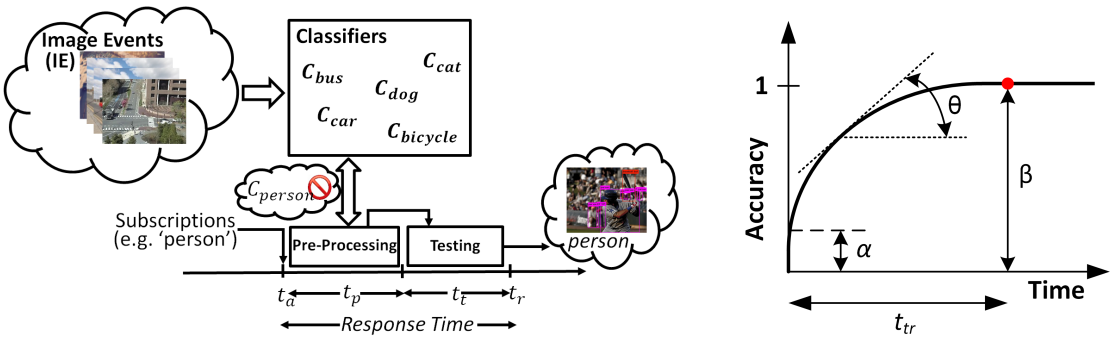
2.6 Response Time Problem Formulation

Consider an example of multimedia event processing shown in Fig. 2.3(a). Suppose a user subscribes at the time "t_a" for the detection of "person" in a stream of "Image Events (IE)". The available classifiers (C_{bus}, C_{car}, C_{dog}, C_{cat}, and $C_{bicycle}$) in the multimedia system can only detect bus, car, dog, cat, and bicycle. Thus the proposed model must be directed towards the pre-processing step, which may include *training* the *person* classifier (C_{person}) before *testing* an image event. Detected events at time "t_r" will be ready to propagate to notify users according to the registered subscription. By assuming pre-processing time as t_p, and testing time as t_t, we can formally define response time (t_{rt}) as:

$$t_{rt} = t_p + t_t \tag{2.1}$$

However, the pre-processing stage of multimedia event processing system may include the following two cases:

[1] https://www.flickr.com/services/api/
[2] https://github.com/tzutalin/ImageNet_Utils



(a) Response Time in Multimedia Event Processing (b) Accuracy with Time

FIGURE 2.3: Definition of Response-Time

Case 1 "Classifier for subscription available": Suppose a previously seen (familiar) concept arrived (like car, dog, bicycle, bus, etc.), the cost of training will be eliminated from the pre-processing stage, resulting in $t_p \rightarrow 0$. Thus, response time for seen concepts is:

$$t_{rt} = t_t \tag{2.2}$$

Here response-time depends on the testing (inference) time of object detection models (excluding networking delays). To make classifiers available for seen concepts, base classifiers are constructed using Pascal VOC and Microsoft COCO [26, 27].

Case 2 "Classifier for subscription not available": Assuming the classifier that can identify concepts (like person, truck, traffic_light, etc.) is not available, then by using the similarity of the new (unseen) subscriptions with existing classifiers, we can further classify the present case into the following two scenarios:

(a) Subscriptions require classifiers similar to base classifiers: Consider an example of concept "truck", classifier for *truck* can be constructed from "bus" classifier by domain adaptation. Thus, *pre-processing* time $t_p = t_{da} + t_{dc}$ where t_{da} refers to the time for domain adaptation and t_{dc} refers to training data construction time, and consequently response time as:

$$t_{rt} = t_{da} + t_{dc} + t_t \tag{2.3}$$

Other than base classifiers, we also consider pre-trained models (like ImageNet [67]). For instance, classifiers for concepts like *person*, *truck*, *traffic_light*, *etc.*, can also be generated using the transfer learning technique of "fine-tuning" pre-trained models. As both cases involve domain adaptation, $t_{da} = t_{freeze/fine-tune}$ in the overall response time.

(b) Subscriptions require classifiers completely different from base classifiers: In this scenario, the pre-processing stage necessarily includes training from scratch and data

construction time (t_{dc}). Thus $t_p = t_{scratch} + t_{dc}$ and accordingly:

$$t_{rt} = t_{scratch} + t_{dc} + t_t \tag{2.4}$$

Please note here we are considering only supervised learning, and therefore the data construction time has to be added to pre-processing time prior to the training of classifiers, which could be removed in the future by incorporating unsupervised/semi-supervised approaches.

In this work, we focus on minimizing the response time while having high accuracy. Therefore we need to evaluate the training time (t_{tr}) *i.e.* t_{da} or $t_{scratch}$ and testing time (t_t) in terms of accuracy (a), due to their existing trade-offs [22]. Using the speed/accuracy trade-off, we can obtain accuracy (a) as:

$$a = f(t_{rt}) = f_1(t_{tr}) + f_2(t_t) + c \tag{2.5}$$

where c is a constant, which could be different for equations 2.2, 2.3, and 2.4. Now our aim is to investigate the maximum value of a that can be provided, by finding $max(f_1(t_{tr}))$ and $max(f_2(t_t))$, on minimizing t_{rt} using $min(t_{tr})$ and $min(t_t)$. In particular, $f_2(t_t)$ depends on the testing time of the multimedia processing (presently object detection) model. However, for the determination of $f_1(t_{tr})$, trends of accuracy have been analyzed using the timeliness for training time (t_{tr}) for all of the identified cases of domain adaptation and training from scratch (t_{da} and $t_{scratch}$) respectively, with various parameters detailed in Fig. 2.3(b). Please note here (in Fig. 2.3(b)) α corresponds to the initial accuracy achieved by a classifier before training, β is the highest accuracy that a classifier can achieve after training, and θ is the higher slope.

2.7 Research Questions

RQ 1: How can we answer multimedia event based queries online consisting of seen concepts of any domain while achieving high accuracy and minimizing the response time?

RQ 2: How can we answer multimedia event based queries online consisting of completely unseen subscriptions (unbounded vocabulary), using an adaptive classifier construction approach with the tuning of hyper-parameters while achieving high accuracy and minimizing the response time?

RQ 3: **(a)** How can we answer multimedia event based queries online consisting of unseen subscriptions (unbounded vocabulary), using domain adaptive classifier construction approach with knowledge transfer from seen subscriptions (bounded vocabulary) while achieving high accuracy and minimizing the response time?

(b) How can we answer multimedia event based queries online consisting of unseen subscriptions (unbounded vocabulary), using task as well as visual domain adaptive classifier construction approach with knowledge transfer from seen subscriptions (bounded vocabulary) while eliminating the requirement of bounding box annotations availability, achieving high accuracy, and minimizing the response time?

It is worth noting that the research hypotheses (RH) presented in Section–1.6 are associated with the above research questions. Explicitly, RQ 1 is related to RH-I, RQ2 is associated with RH-II, RQ3 (a) with RH-III, and RQ3 (b) with RH-IV.

2.8 Summary

This chapter is designated to the detailed discussion of the problem domain and its technical limitations. Debate on motivation also states the need for generalizable multimedia event processing followed by the requirements for the comparative analysis of the related work. Challenges and opportunities for the generalized multimedia event processing-based applications using IoMT by taking object detection as a case study are also highlighted. I also formulate the concept of "response-time" that I use throughout the problem and its proposed approaches. Finally, the formulated problem is divided into three research questions which I tackle in different chapters of this thesis. However, before the proposed solutions in other chapters, I give a state of the art of IoMT and a detailed analysis of object detection in the next Chapter–3.

Chapter 3

Background and Related Work

3.1 Introduction

In this chapter, first, I discuss the literature of the Internet of Things (IoT) for investigating the concept of the Internet of Multimedia Things (IoMT) in Section–3.2. I define the IoMT and presents visions of IoMT in light of IoT. A comparison of IoT and IoMT is also provided based on the characteristics and describes requirements with existing solutions of IoMT. Section–3.3.1 analyzes the current deep neural network-based object detection models, which I use in this thesis work. A detailed comparison of object detection datasets is presented in Section–3.3.2 which I use to train classifiers for proposed models discussed in Chapters–5 to 8. Comparison of object detection models and datasets demonstrate the need to bridge their large gap of performance and extensive vocabulary.

Some of this related work, along with limitations and future directions, have been presented in the journal titled "A Survey on Object Detection for the Internet of Multimedia Things (IoMT) using Deep Learning and Event-based Middleware: Approaches, Challenges, and Future Directions [31]" of Image and Vision Computing (IMAVIS), Elsevier.

3.2 Multimedia in Internet of Things

This section discusses the concept of the Internet of Things (IoT) and the Internet of Multimedia Things (IoMT), along with its characteristics, challenges, and existing solutions for the processing of multimedia.

3.2.1 Concepts of IoT & IoMT

3.2.1.1 Internet of Things

The European Research Cluster of IoT (IERC) [68] definition states that IoT is "A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network.". Semantically IoT stands for "a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols", where things could be RFID) tags, sensors, everyday objects, actuators, smart items like mobile phones etc [69]. Basically, IoT allows people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/network and Any service [48, 70, 71].

The three visions of IoT i.e. "Internet oriented", "Things oriented", and "Semantic oriented", are represented in literature [2] as in Fig. 3.1(a). "Things Oriented" perspective of IoT first considers Radio-Frequency IDentification (RFID) tags in its definition. Similarly, Unique/Universal/Ubiquitous IDentifier (uID) [72] are also part of IoT vision which is much broader than only object identification. Near Field Communications (NFC) and Wireless Sensor and Actuators are also responsible for the build-up of the IoT [73]. Projects like Wireless Identification and Sensing Platforms (WISP) also developed to provide appropriate platforms for IoT. Also, it reports that traffic generated/received by everyday objects in IoT will overcome the traffic caused by the networking of humans [74]. Another concept, namely *spime*, also emerges as an object that could be tracked through space and time throughout its lifetime, and that will be sustainable, enhanceable, and uniquely identifiable [75]. However, spime is almost similar to *Smart Items*, which also consists of wireless communication, memory, and elaboration capabilities [2]. The next technological revolution (connecting people anytime, anywhere) is to connect inanimate objects a communication network appear in ITU Internet Report [76], and thus we are connecting the world of people with the world of things i.e. connectivity for anything. By considering the functionality and identity as central, IoT appeared as "Things" having identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environmental, and user contexts [69].

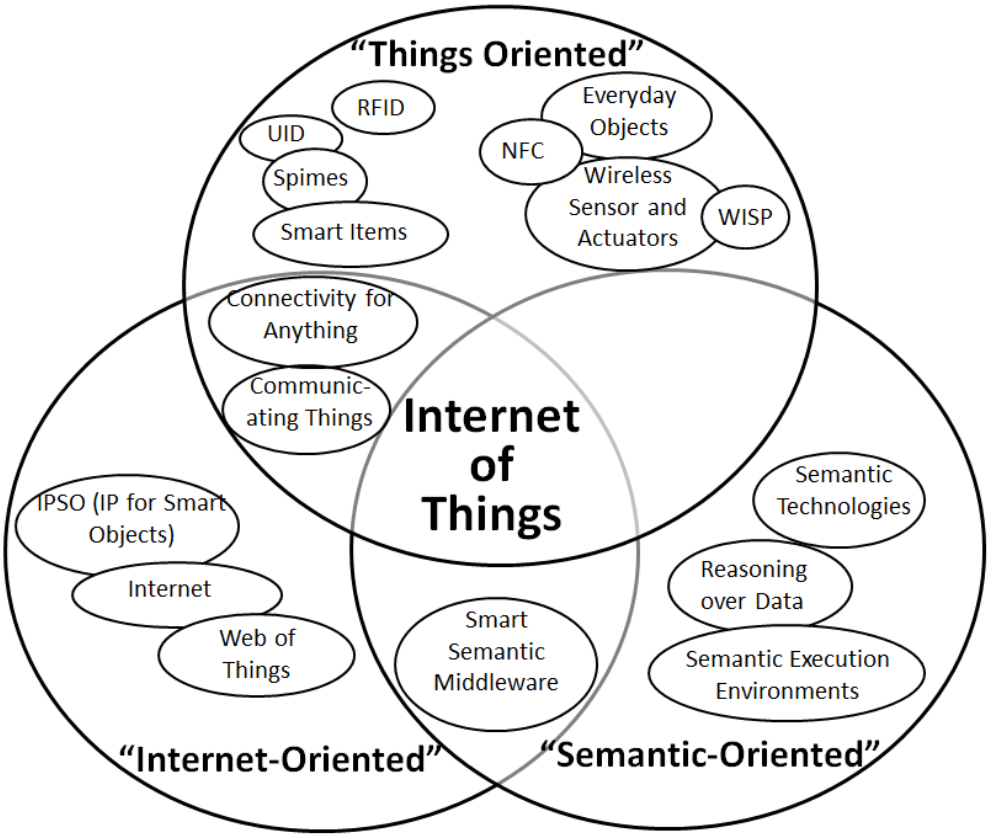
Internet Protocol also got promoted for connecting Smart Objects around the world for the IoT vision of the IPSO (IP for Smart Objects) [77]. "Internet-0" [78], a new kind of network of everyday devices, extends the original notion of internetworking to inter-device internetworking, and thus also agreed to make those devices to intercommunicate

and interoperate from any location. Integrating real-world devices to the web ("Web of Things", another vision of IoT) is also beneficial for devices to interact in the same language as other resources on the Internet, and thus making it relatively easy to integrate physical devices with any other Web page [79]. "Semantic Oriented" visions of IoT presented in research work [2, 80–83] directed to address the issues arising due to the increase in heterogeneity of devices involved in the future Internet. Semantic execution environments, semantic technologies, reasoning over data, smart semantic middleware, and dataspaces [84, 85] consist of various solutions that can fulfill many requirements of IoT, including representation, storing, scalability, communication, search, organized information, etc.

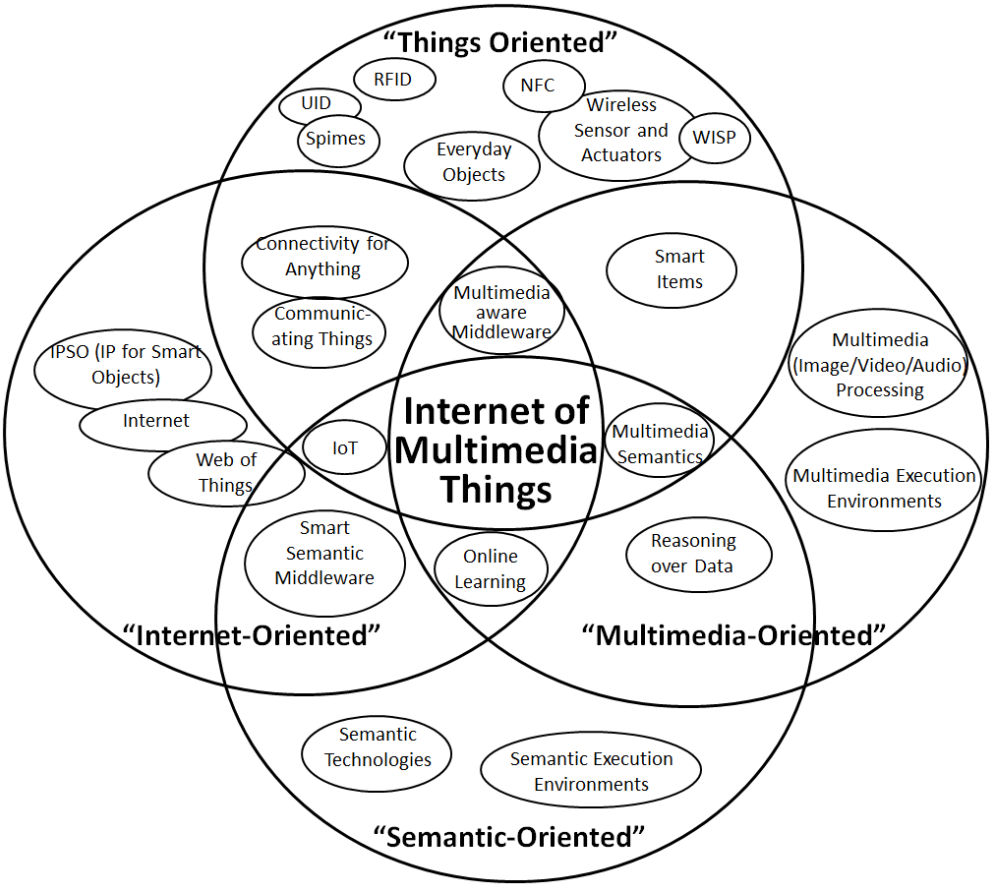
3.2.1.2 Internet of Multimedia Things

There are numerous applications of IoT belonging to multiple domains such as medical, transportation, security, business transactions, retail, agriculture, monitoring, process automation, personal and social domains, etc. Although most of the real-time data of these domains consist of images/videos, the challenges of processing multimedia data are yet to handle in IoT. This requirement of enabling objects of smart cities with the ability to observe, sense, and understand each other, compelled the research to move from conventional IoT to multimedia-based IoT [7, 86–90]. As the concept of integrating multimedia with IoT is very recent, it has been referred to in the literature with IoMT and MIoT synonymously and needs standardization [5, 91]. In literature [86, 92, 93], IoMT realized as the addition of challenges over IoT, which may include Security, Routing, Quality of Service (QoS) and Quality of Experience (QoE) concerns, Heterogeneity of multimedia sensors, etc. I defined the IoMT in this thesis as an IoT-based paradigm that allows objects to connect and exchange structured as well as unstructured data with one another to facilitate multimedia-based services and applications [32].

Visions of IoMT realized in the presented work shown in Fig. 3.1(b) that are adapted from visions of IoT (Fig. 3.1(a)). It is crucial to include "Multimedia Oriented" with three other visions (Things, Internet, and Semantic) of IoT, due to the nature of data which could be semantic (structured) based or multimedia-based. Also, we can see in Fig. 3.1(a), that the overlapping of "semantic-oriented" with "things-oriented" is empty and states the gap in the literature of IoT [2, 94, 95]. Similar to "Smart Semantic middleware", we also have multimedia aware middlewares to provide services of handling multimedia based events between distributed platforms and applications. Existing research in this category is focused on the establishment of architecture [96] for Multimedia Internet of Things while handling the issues of multi-modal Big data computation [4], with scalability and maintainability of the model for effective multimedia information



(a) IoT [2]



(b) IoMT

FIGURE 3.1: Visions of IoT [2] to our visions of IoMT

sharing [6]. Also, smart items are not only part of "things" as their core origin is multimedia (audio/images/video/text). "Multimedia Semantics" [97–99] is another popular domain associated with content semantics, situational semantics, retrieval semantics, and social semantics of multimedia data, where all of them become parts of IoT at some point in the processing of events. The goal of "Reasoning over data" cannot be fully accomplished by using only "semantics"; multimedia understanding and reasoning, have attracted a large amount of interest from the research specifically in machine learning [100–107]. Processing of multimedia events and their execution environments are essential in IoMT for analyzing the huge amount of unstructured data generated in smart cities [108–113]. Online learning is also a well-received domain in the field of smart cities in analyzing and/or processing of semantics as well as multimedia based data [114–118].

3.2.2 Characteristics, Requirements, and Solutions for IoMT

A detailed comparison of IoT and IoMT based technologies is shown in Table 3.1 using service-oriented architecture (SoA) for IoT [2, 94]. Existing technologies of IoT can be easily distinguished from IoMT by analyzing *sensing*, *networking*, *service*, and *application-level* services.

Sensing layer is integrated with tags/sensors, able to automatically sense data, and communicate information among things [94]. In-order to determine existing sensing abilities of IoT as compared to IoMT, the sensing layer divided into characteristics of resources, deployment, heterogeneity, communication, and scalability. It is presented in the work [121], that low-cost handling of multimedia data while preserving the effectiveness is the major challenge for the IoMT based systems as compared to IoT. Conventionally Radio Frequency IDentification (RFID) tags are the basis of wireless sensor networks for tracking of scalar data in IoT. Technological advancement in the field of Radio-Frequency IDentification (RFID) with sensing and actuating capabilities is the key step of IoT that enabled the communication in smart cities without the help of humans [119, 120]. The high power consumption of multimedia (audio and video) sensors as compared to scalar data sensors is one of the key challenges in the deployment of multimedia over IoT [142]. Multimedia content, e.g. audio, video, etc., acquired possess distinct characteristics as compared to the scalar data acquired by typical IoT devices. Scalar data is relatively very less heterogeneous as compared to multimedia data. Scalar sensors are typically designed for monitoring temperature, pressure, humidity, location of objects, or any other measured values, while standard audio and video sensors for capturing sound, still or moving images [5, 122, 124]. Most of the research in the field of the wireless sensor network (WSN) is concerned with scalar sensor networks. Thus the concept of Wireless Multimedia Sensor Networks (WMSNs) is receiving attention in-order

TABLE 3.1: Comparison of IoT and IoMT based systems

Characteristics		**Internet of Things (IoT)**	**Internet of Multimedia Things (IoMT)**
Sensing	Resources	Low Cost, Size, and Energy consumption [119, 120]	Low Cost, Size, and High Energy consumption [121]
	Deployment	RFID tags (one-time or application dependent) [119, 120]	Video and Audio Sensors [5, 122]
	Heterogeneity	Limited Heterogeneity (Scalar data) [5, 122]	Heterogeneous (Multimedia data) [5, 122, 123]
	Communication	WSN-based Protocols (ZigBee, WLAN, Bluetooth, WiFi, UMTS etc.) [94]	WMSN-based Protocols (RMST, PSFQ, ESR, CODA, MRTP etc.) [124] (Non-Standardized and Application Specific)
	Scalability	Highly Energy-Efficient sensors and Coexistence for WSN Protocols [94]	Highly Energy-Efficient sensors and Coexistence for WMSN Protocols (not standardized [124])
Networking	Networks	Topology (multi-hop, mesh or ad hoc) [94]; and Node Operation: Predefined [5]	Topology (multi-hop, mesh or ad hoc) [94]; and Node Operation: Adaptive [5]
	Quality of Service (QoS)	Low delay, packet loss, jitter and Bandwidth [94]	Low delay, packet loss, jitter and High Bandwidth[86, 125–127]
	Storage, Searching and Processing	Data Mining and Analytics [128–131]	Data-mining, Feature Extraction, and Cloud-based Multimedia Storage System [132–134]
	Security and privacy	Data Confidentiality, Privacy, and Trust (TEA, AES, ECC etc. [94])	Confidentiality-Preserving [121], Security and Trust [7, 92]
Services	Service Composition	SOA-based and event-based middleware [135]	No available specialized middleware [7]
	Service Management	RFID-based Service Architectures [136]	Video on Demand (VoD) service, MVSWN, HIVE etc. [137]
	Service APIs	Universal API [94] and Effective Domain Specific Services (like Health Thermometer Service [94])	WiSNAP and AER [133, 138, 139]

Applications	Infrastructure & Industrial	Transportation, business transactions, online-payment, smart cities, environmental monitoring, and smart homes and building [2, 94]	Traffic Monitoring, Airport Surveillance, Municipality Supervision, Behavioral interpretation systems [5, 137, 140]
	Health-care applications	Triage, patient monitoring, personnel monitoring, disease spread modelling and containment [120]	Medical Imaging, Telemedicine [5, 141]
	Security	Tracking, Losses, Identification and authentication [2]	Automated Public Security, Building Security, Airport Security, Surveillance, Crowd Monitoring [5, 11]
	Personal and social domain	Facebook, Twitter, Google Calendar, Loss and Stolen objects notification [2]	Multihoming [87]

to enable technologies for multimedia content. High energy-efficient sensors with the coexistence of communication protocols (based on WSN and WMSN) are necessary for the scalability of IoT and IoMT. IoT support numerous protocols (like ZigBee, WLAN, Bluetooth, WiFi, UMTS, etc.) for different communications. However WMSNs based protocols (like RMST, PSFQ, ESR, CODA, MRTP, etc.) are only domain-specific and not standardized yet which also associate challenges with communication and scalability of multimedia applications [5, 94, 124].

Networking layer [94] is responsible for providing infrastructure that allows things to connect over wireless or wired networks, which is crucial for the sharing of IoT/IoMT based data (scalar/multimedia). Amongst the characteristics of the network layer, adaptive node operation is the main difference of the IoMT based networks as compared to the networks for only IoT [5]. IoMT follows the same characteristics as of IoT in terms of topology, low delay, packet loss, jitter, etc. However, the fixed bandwidth of IoT based systems, which is usually low due to the expectation of small size packets, is also not sufficient to provide requisite Quality of Service (QoS) [126] for IoMT. Most of the storage, searching, and processing techniques are available only for IoT based systems, and very few like data mining, feature extraction, cloud-based multimedia designed for IoMT [132–134]. Similarly, loads of security and privacy based protocols (like (TEA, AES, ECC, etc.) are available for IoT, but only a few are handling the challenges of IoMT [7, 92, 121]. A novel security-critical multimedia service architecture proposed in the work [92]. It also contributes towards analyzing and classifying traffic classification for

various multimedia streaming applications to illustrate the effectiveness of the proposed model. Another low-cost data acquisition and confidentiality preserving framework [121] proposed for IoMT. It is based on two-layer security protection *i.e.* chaotic encryption control during the sampling process and chaotic permutation-diffusion encryption after the sampling, both of these encryption operations have low computational complexity. In a few recent approaches, layered based protocols have been presented for handling the challenges of QoE associated with the concept of Multimedia IoT (MIoT) and analyzed using IoT based vehicle application [86, 127, 137]. Similarly, challenges related to compressed sensing video streaming [126] and efficient cloud-based transmissions [134] including robust multicast routing [143] realized due to high bandwidth requirements of multimedia applications in the Internet of Things.

The service layer designs to enable services and applications in IoT [2, 94]. This layer relies mostly on *middleware* technologies in-order to support service providers and users, which abstracts the complexities of the system/hardware. Service Oriented Architecture (SOA) recognized as a good solution for the IoT middleware [48, 119]. SOA- and Event-based middleware are designed specifically for handling the structured data, and no specialized middleware available for the service composition of IoMT [7, 135]. Research work [136] presented in highlights the benefits of Service-Oriented Computing (SOC) to construct middleware for the Internet of Things. They proposed a Radio-frequency identification (RFID) suite (middleware), designed on a multi-layer architecture while leveraging SOC. Here the role of the middleware is to track RFID-tagged objects as well as other objects that can provide relevant information. A comprehensive review of video streaming also presented in paper [137] mainly focuses on vehicular communication perspective in Multimedia oriented Internet of Things (IoT) environments. By emphasizing the growth of multimedia traffic on the overall Internet, this review discussed many of the services particularly related to "vehicles to vehicle" or "vehicles to IoT devices". In terms of Service APIs, a large number of universal, as well as domain-specific APIs, can be found in reviews of IoT [2, 94, 119, 120]. Ongoing research on prototypes of multimedia sensors and their integration into testbed described in paper [133]. It indicates Address event image sensing (AER) and Wireless Image Sensor Network Application Platform (WiSNAP) as software and application programming interface for multimedia based networks. AER is a software tool to identify the occurrence of an event without sending back real images. All sensors in AER visualized as nodes of a neural network and camera as a detection tool by the node. The binary decision of nodes used to detect event patterns by the AER tool [138]. WiSNAP presents an application interface to image sensors, which is a first towards ease of use for multimedia-based communications. Its framework consists of two sub-parts *i.e.* the image sensor API and wireless mote API [139].

Multimedia streaming (especially videos) is one of the most common types of events within the applications of smart cities [144]. Its applications may include health-care, emergency services, crowd monitoring, traffic management, building management, smart environment, personal and social domains. However, multimedia-based systems [5, 11, 87, 137, 140, 141] designed for the applications of smart cities are still comparable to the systems designed for processing structured data [2, 94, 120], and need to be further investigated using IoT.

3.3 Object Detection

3.3.1 Deep Neural Network based Object Detection Models

Deep convolutional neural networks are proven to be suitable for image recognition in achieving high-performance results. Thus we have compared the most recent and competitive object detection model (Faster RCNN, SSD, YOLOv3, and RetinaNet) that could prove to be prominent for the multimedia event processing. Neural Network-based architectures for all object detection models shown in Fig. 3.2 and explained below in detail.

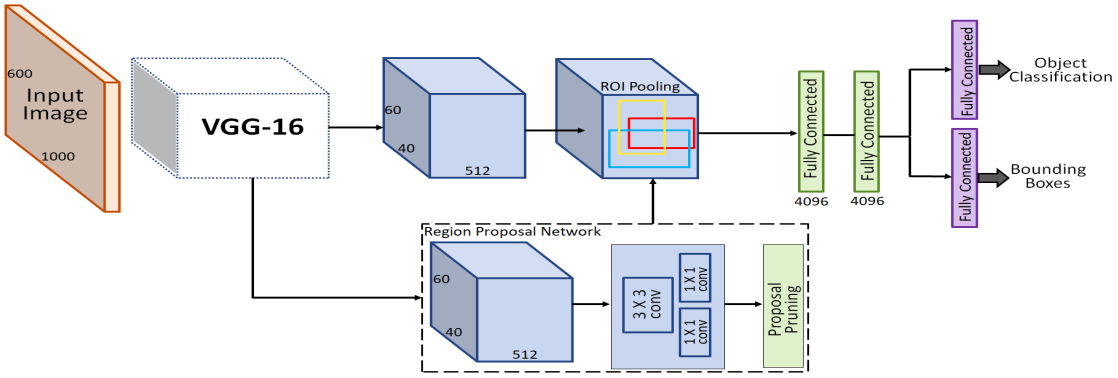
Faster R-CNN The R-CNN (Region-based Convolutional Network) [145] model was among the first model to use convolutional neural networks for the detection of objects. These models begin with the first region search and then perform the classification. R-CNN uses a *selective search* method [146] in-order to create bounding boxes or region proposals, and also combine deep learning to identify objects in these regions. The aim of the Fast Region-based Convolutional Network (Fast R-CNN) [147] is to reduce the time complexity by removing the need for feeding the high number of region proposals to the convolutional neural network every time. In this case, the convolution operation is done only once, by taking an entire image and a feature map is generated from it. However, the use of the selective search method for the detection of region proposals is still necessary for the Fast RCNN model, which is considered computationally expensive. Faster RCNN [148] removes the need for using a *selective search* algorithm by proposing a separate network to predict the region proposals, known as Region Proposal Network (RPN). RPN accelerates the training and testing with improvement in performance. The predicted region proposals are then reshaped, classify the image, and finally generates an output for rectangular bounding boxes.

Mainly, the Faster R-CNN model utilizes RPN and the Fast R-CNN model, for the detection of objects. The neural network-based architecture of Faster RCNN is shown

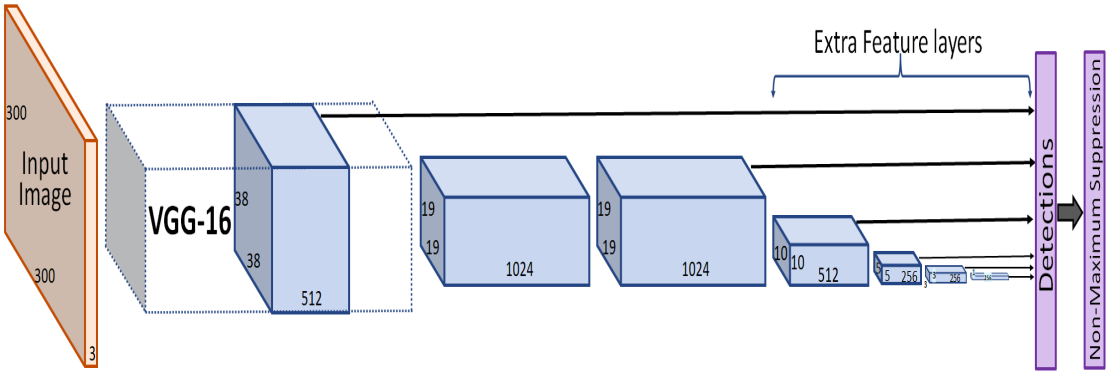
in Fig. 3.2(a). First, the Faster R-CNN extract feature maps from Input Image (resized to 1000×600) using the backbone convolutional neural network (presently VGG-16 [18]). RPN check locations that contain an object and pass bounding boxes to the detection network. RPN uses a sliding window to find each location on the input image by placing a set of *anchors* on the output feature map of the backbone network. For instance, 9 anchor boxes are used with 3 scales (128, 256, and 512) and 3 aspect ratios (1:1, 1:2, and 2:1) in Pascal challenges by Faster R-CNN. As proposed regions could by highly overlapping, NMS (Non-Maximum Suppression) is used by Faster-RCNN to reduce the number of region proposals. Other than RPN, the detection network of Faster-RCNN is similar to Fast R-CNN. It also consists of a backbone, ROI pooling layer, two fully connected layers followed by two fully connected branches for the object classification, and bounding boxes regression. ROI pooling layer also uses the feature map generated by the backbone network. ROI pooling layer considers the regions corresponding to the bounding boxes proposals generated by RPN. It divides the regions into a fixed number of windows and performs maximum pooling for the fixed output size. Then two fully connected layers take the output of the ROI pooling layer, and features are passed object classification and bounding boxes regression layers. Here, the classification layer also makes use of the softmax layer to get the classification scores. The regression layer helps in the improvement of the predicted bounding boxes.

Faster R-CNN was 10 times faster than the other R-CNN models and designed to achieve high accuracies in less time. However, it ends up being the slowest than other object detection models (like SSD, YOLO, RetinaNet, etc. discussed below) and not the most accurate.

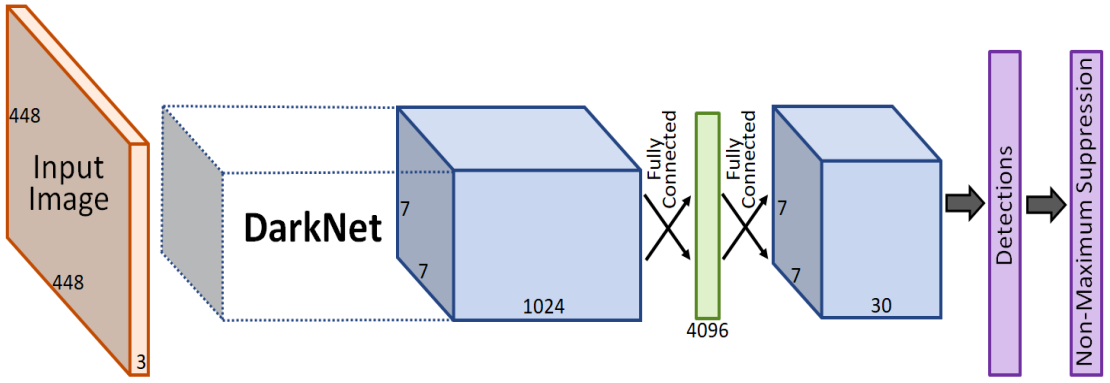
Single Shot Detection (SSD) The Single Shot Detector (SSD) framework [36] realizes the requirement of real-time applications and focused on high speed while maintaining accuracy. SSD mainly consists of two parts: a backbone network and SSD head. Backbone is the same as the standard feature extraction network (like base network VGG-16 [18] used in Faster R-CNN), which is pre-trained on image classification dataset (like ImageNet). SSD head consists of extra layers to produce detections with additional features, namely multi-scale feature maps for detection (to allow predictions of detections at multiple scales), convolutional predictors for detection (where each added feature layer produce fixed set of predictions using a set of convolutional filters), and default boxes and aspect ratios (SSD use anchor boxes at various aspect ratio and learns the off-set rather than learning the bounding box). Fig. 3.2(b) shows the feed-forward convolutional network of SSD, the first few layers are the backbone, and extra feature layers represent the SSD head.



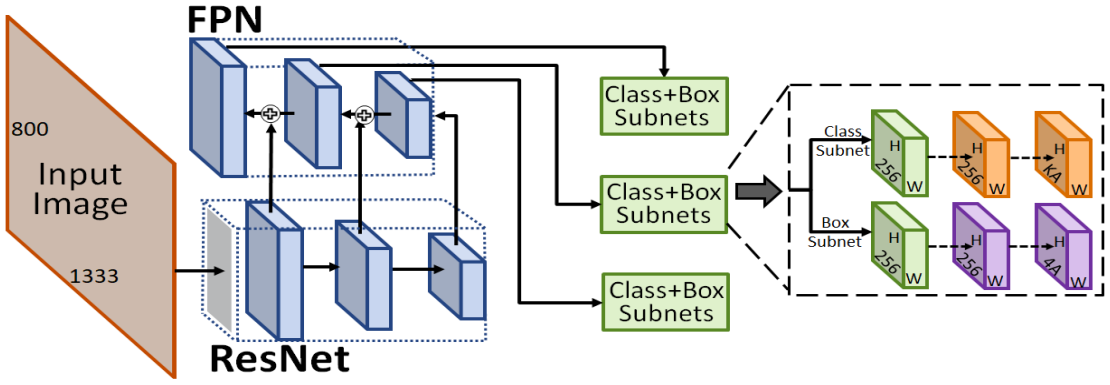
(a) Faster RCNN



(b) SSD



(c) YOLO



(d) RetinaNet

FIGURE 3.2: Neural Network based Architectures of Object Detection Models.

SSD does not make use of Region Proposal Networks (RPN) and predict objects directly from feature maps. It resolves the need for RPN by using small convolution filters (like 3×3) to predict bounding boxes and class scores. First, SSD extracts feature maps from Input Image (300×300) using backbone VGG-16. Then SSD uses the rest of the convolutional layers to detect objects. It utilizes SSD six more auxiliary convolution layers after the VGG16 for predictions. SSD predicts 21 scores (in case of Pascal VOC Challenge) where class "0" indicates no objects and 1 boundary box for each class. Another additional feature of SSD from Faster R-CNN is to use default boundary boxes. During the initial training, models could be very unstable, thus instead of making random guesses of boundary boxes while training, the more sensible approach is to use the ground truth as default boxes for the guesses. However, these default boundary boxes are selected manually by the SSD model. Finally, SSD also uses the non-maximum suppression step to remove duplicates and produce the final detections.

In general, SSD addresses the speed-accuracy trade-off and achieves a good balance between them, which makes it suitable for real-time processing applications.

You Only Look Once (YOLO) You Only Look Once (YOLO) [149] is a single neural network to predicts bounding boxes and class probabilities directly from an image in a single evaluation. YOLO sees the whole image at once as opposed to the previous region-based object detection models, which only look at generated region proposals, which helps YOLO in avoiding false positives. YOLO views image detection as a regression problem that divides the entire image into a grid of S×S, and each grid predicts B bounding boxes and confidence. YOLO also does not require the step of region proposals and thus predicts bounding boxes in real-time.

Unlike conventional object detectors, YOLO uses *darknet* as a backbone network (pretrained on ImageNet Classification data), its network architecture is shown in Fig. 3.2(c). YOLO network consists of 24 convolutional layers followed by 2 fully connected layers. Here few convolution layers use 1×1 filters followed by 3×3 filters. Darknet, produces the output it outputs a tensor with shape (7×7×1024) tensors, and then after 2 fully connected layers, it outputs (7×7×30) shape tensor of predictions. Please note that if one image contains S×S×B bounding boxes, then final prediction tensor values for one image is S×S×(5B+K), where each box may consist of 5 outputs: 4 predicted locations, 1 confidence score, and K (=20 classes for PASCAL VOC Challenge) conditional probabilities.

YOLO is very fast and can detect objects in real-time [149]. However, the limitation of the YOLO object detection model is that it struggles in predicting small objects, and it is still falling behind in terms of accuracy, from state-of-the-art detection models.

RetinaNet It is the assumption in the RetinaNet object detection model [37] that the foreground-background class imbalance problem is the cause of the inferior performance of one-stage detectors as compared to two-stage detectors. Thus RetinaNet model introduced a new loss function "Focal Loss" to deal with class imbalance. Here in the first stage, the classifier applies to a sparse set of candidate object locations. The second stage is responsible for classifying the location of each candidate as one of the foreground classes or as background. It first reshapes the entropy loss so that it lowers loss weights assigned to easy negative samples. Thus "Focal loss" focuses training on a sparse set of hard examples, which improves prediction accuracy.

RetinaNet network (shown in Fig. 3.2(d)) mainly consist of: ResNet+FPN backbone, object classification subnetwork, and bounding boxes regression subnetwork. FPN (Feature Pyramid Net) is used on top of ResNet to construct the feature map from Input Image (800×1333). Classification subnet is a fully convolutional network consist of four 3×3 convolutional layers with 256 filters, then one 3×3 with K×A filters. The shape of its output is (W, H, K, A), where W×H are dimensions of the feature map, K is the number of classes, and A represents anchors. Like classification subnet, the bounding boxes regression subnet is also connected to FPN and output similar shape with the exception of 4A filters in the last 3×3 convolutional layers. Finally, top predictions get merge from all levels, and RetinaNet produces predictions.

RetinaNet is efficient and accurate than the previous region-based convolutional networks while using ResNet-101-FPN [51] as the backbone for feature extraction.

Comparison of DNN based Object Detection Models: From the perspective of detection of objects in processing multimedia events of smart cities, we have analyzed deep neural network based models using the following dimensions:

- Backbone: It represents the backbone used for the step of image classification in the object detection model. For instance Inception-v1, -v2, -v3, Resnet-101, Mobilenet-v1, and -v2 [51, 150, 151] are some of the examples of backbones. Most of the time each object detection model have one recommended backbone, but they are also flexible to classify image using other backbones. However, models itself provide a validated performance on the use of a few other backbones, while the use of any completely different backbone is the responsibility of the designer.
- Mean Average Precision (mAP): It is the mean of average precision calculated on the detection of each class on which the model is trained. It is among one of the crucial matrices which need to be evaluated before proceeding towards choosing any object detection model for accurate multimedia event processing.

TABLE 3.2: Comparison of DNN based Object Detection Models

Model	**Backbone**	**mAP**	**FPS**	**Training Time (on GPU)**	**Classifier Size**
YOLOv3	Darknet53 448x448 (Also Flexible with AlexNet, VGG-16, Extraction, Darknet19, Darknet19 448x448, Resnet-18, -34, -50, -101, -152, ResNeXt-50, -101 (32x4d), -152 (32x4d), Densenet 201 and Darknet53)	33.0	45fps	13 to 30 hours	248.0MB
RetinaNet	Resnet101 (Also Flexible with Mobilenet-128, -160, -192, -224, Resnet-50,-152, VGG-16, -19, Densenet-121,-169, and -201)	37.8	5fps	10 to 35 hours	152.7MB
SSD	VGG-16 (Also Flexible with ResNet-101)	28.8	19fps	20 - 40 hours	137.3MB
Faster R-CNN	VGG-16 (Also Flexible with ZF-Net-16 and Resnet-50)	27.2	2fps	16 to 84 hours	548.3MB

- FPS: Frame Per Second (FPS) denotes the time taken by model to evaluate an image (frame) for the detection of objects.
- Training Time: Covers the time required by model to train a classifier for a single class until it reaches to maximum accuracy. It is necessary to add the training time to total inference (evaluation) time of one frame, for the computation of total delay of object detection model before deploying it in real-time applications.
- Classifier Size: The space complexity of model in order to store the classifier after training.

Table 3.2 represents a comparison of these object detection models. It represents recommended backbones of these models, with the flexibility of using others, there mean Average Precision (mAP) on using classes of Pascal VOC, Microsoft COCO, and Open Images dataset. YOLOv3 has the highest frames per second (i.e. number of images processed per second) with competitive accuracy. However, classifier sizes could be a major constraint in the deployment of such models, and SSD or RetinaNet have relatively minimum classifier sizes. Unfortunately, Faster-RCNN could not be suitable for real-time applications of smart cities. Thus we can conclude these object detection models have their trade-offs in terms of speed, accuracy, memory, and required training time,

and need to be used optimistically for online training to achieve low response time for multimedia based application of smart cities.

Deep Neural Network based Object Detection Models: These models cover the most recent object detection models and famous for their high accuracy and speed of processing image events. Though training time is not considered a comparison metric in these models, such models are adaptable for any new domain if training data is available. However, such models themselves don't support any vocabulary beforehand and need to utilize the object detection datasets or any domain-specific datasets for their applications.

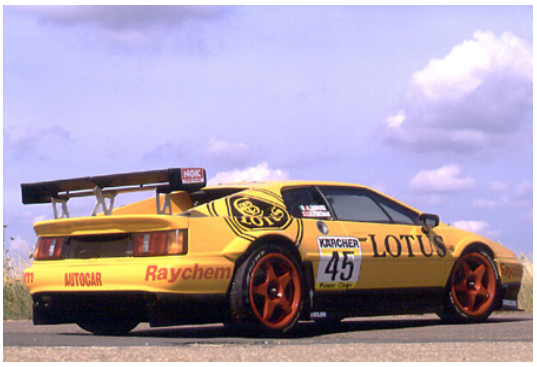
3.3.2 Seen Classes based Object Detection Datasets

Technical advancements of machine learning models could be beneficial only with the availability of annotated datasets. Among large number of popular datasets of machine laerning models, we have analyzed four visual datasets, namely ImageNet [152], Pascal VOC [26], MSCOCO [153], and OID [154], which are globally considered suitable for training of object detection models.

Image_Net: The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [152] has been running annually from 2010 to present and become the standard benchmark for large-scale object recognition (Image Classification as well as Object Detection). ImageNet is an image dataset organized according to the WordNet hierarchy [29, 155]. ILSVRC annotations follow one of the two categories *i.e.* (1) image-level annotation for the presence or absence of an object class in the image (by taking a binary decision) (2) object-level annotation for the detection of bounding box of an object and its class label. It consists of more than 1000 object classes having around $1,461,406$ images. However, 200 basic-level categories are only available for the testing of the object detection task.

Pascal VOC: The PASCAL Visual Object Classes (VOC) [26] is another publicly available dataset of annotated images, designed specifically for object detection. Like ImageNet, this challenge also consists of two components, i.e. image dataset with ground truth annotation of images and an annual competition with a workshop where images are obtained from the Flickr website[1]. It mainly consists of 20 classes *i.e.* aeroplane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, and TV. It also attempts to address three principal

[1]Image hosting service (flickr.com)



(a) ImageNet

(b) Pascal VOC

(c) Microsoft COCO

(d) Open Images Dataset

FIGURE 3.3: Example Images for Object Detection Datasets

challenges, namely classification (does the image contain any instances of a particular object class?), detection (where are the instances of a particular object class in the image (if any)?), and segmentation (to which class does each pixel belong?). The total number of images and objects in the main datasets is $11,540$ and $31,561$, respectively. Also, the total number of images and objects in segmentation datasets is $2,913$ and $6,934$ respectively. However, the number of objects of a particular class in this dataset varies drastically, like class *person* have 10,129 objects for training and validation while class *cow* has 702 objects only. At the same time class *dog* has 1,541 objects.

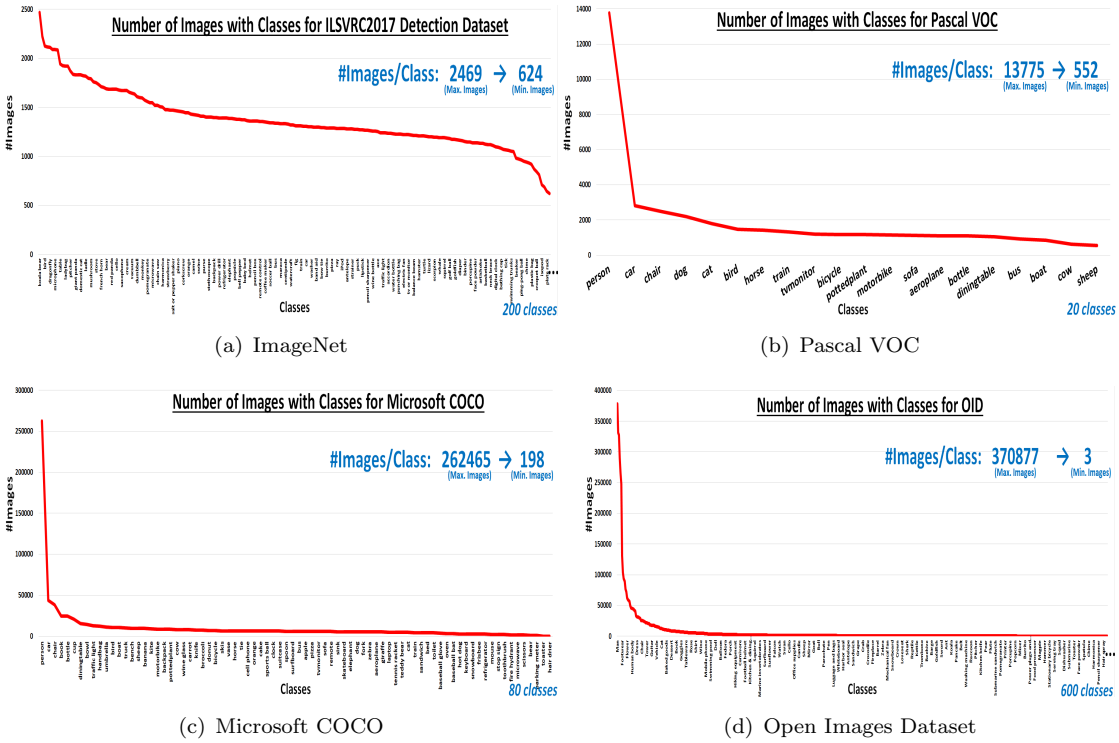
Microsoft COCO: The Microsoft Common Objects in the COntext (MSCOCO) dataset [153] introduced a new large-scale dataset that addresses three core research problems: detecting non-iconic views of objects, contextual reasoning between objects, and the precise 2D localization of objects. The creation of this dataset depends on the extensive involvement of crowd workers via different user interfaces for category detection, instance spotting, and segmentation. MSCOCO consists of 91 common object categories with 82 of them having more than $5,000$ labelled instances. Here a selection of 91 categories is based on picking categories of high votes and keeping a balance among the number of categories per super-category (person, vehicle, outdoor, animal, accessory,

sports, kitchen, food, furniture, electronic, appliance, and indoor). In total, the dataset has 2.5 million labelled instances in 328k images. The ratio of object instances per image for MSCOCO is 7.7, which is considerably more as compared to other datasets (ImageNet (1.1) and PASCAL VOC (2.4)).

Open Images Dataset Dataset Open Images V4 [154] is a collection of 9.2 million annotated images available for image classification, object detection, and visual relationships. Specifically open images V4 is large scale in terms of images ($9,178,275$), annotations ($30,113,078$ image-level labels, $15,440,132$ bounding boxes, $374,768$ visual relationship triplets) and the number of visual concepts (classes) ($19,794$ for image-level labels and 600 for bounding boxes). Particularly for object detection, this distribution can be represented as 15.4 million bounding boxes for 600 categories on 1.9 million images. Moreover, annotations of OID images are rich enough to have an average of 8 annotated bounding boxes per image, which guarantees its suitability for the detection of objects. Its image acquisition procedure mainly includes identification of all Flickr[2] images with CC-BY(Creative Commons Attribution) license, downloading original images, extract relevant metadata, removing common/inappropriate/duplicate images, and finally partitioning of images into training ($9,011,219$ images), validation ($41,620$) and testing ($125,436$) datasets. After that, OID has shortlisted 600 object classes, classify them with the help of image classifiers and humans, then generate bounding boxes using reasonable guidelines (details appear in with up-to-date dataset on Open Images V4 website[3]).

Fig. 3.3 represents an example of images in these object detection datasets. It can be observed that OID and Microsoft COCO dataset consist of real-world images having more number of classes. Pascal VOC dataset also includes smart city scenes, with more instances but having less number of classes. However, ImageNet focuses on image classification and thus having iconic images for the detection of objects also. My investigation (Fig. 3.4) shows that existing object detection datasets designed for real-world images of smart cities do not have enough classes [26, 28, 152, 153]. Moreover, the datasets that claim to have more number of classes have a fewer number of images for most of the classes. Thus resulting object detection datasets are less accurate. The datasets that perform best and have high mean average accuracy have the least number of classes. It can be concluded that this trade-off of performance with the number of classes is perceived and not addressed by any object detection datasets.

[2]Image hosting service (flickr.com)
[3]https://storage.googleapis.com/openimages/web/index.html



(a) ImageNet (b) Pascal VOC

(c) Microsoft COCO (d) Open Images Dataset

FIGURE 3.4: Number of Images Annotated in Object Detection Datasets with classes.

Comparison of Object Detection based Datasets From the perspective of detection of objects in processing multimedia events of smart cities, we have analyzed datasets using following dimensions:

- Number of Classes: Represents the total number of categories (like cat, dog, car, tree, bike etc.) for which dataset have training, validation, and testing images. It is important to note that number of images of particular class may vary from one class to another class.
- Average Number of Objects per Image: It is designated to compute the average number of classes present in one image, as it is a very important factor in the training of any object detection model.
- Average Image Size: It represents the space complexity of dataset for storing images of particular class. However high resolution images are good for training but may take more space and training time. Thus optimal size should be recommended before choosing any dataset for object detection in the real time event processing.
- Average Number of Training Images per Class: It computes the median by using total number of training images available in each class. Since number of images in particular class may vary from one class to another, it is necessary to evaluate this dimension before considering any dataset good for training on all classes.

TABLE 3.3: Comparison of Available Object Detection Datasets

Dataset	**Classes**	**Av. #Training Images/-Class**	**Av. #Validation Images/-Class**	**Av. #Testing Images/-Class**	**Av. #Objects/ Image**	**Av. Image Size**
ImageNet	200	823	58	201	1.1	∼4.6MB
Pascal VOC	20	1170 (train+val)		147	2.4	∼0.1MB
Microsoft COCO	80	4,452	46	370	7.7	∼0.08MB
Open Images Dataset	600	740	26	77	8.1	∼3MB

- Average Number of Validation Images per Class: Similarly it determine the median by using total number of validation images available in each class.
- Average Number of Testing Images per Class: Lastly it evaluates the median by using total number of testing images available in each class.

Table 3.3 represents a comparison of available datasets along the identified dimensions. These dimensions are crucial to analyze before choosing any dataset for the processing of multimedia events in smart cities. It can be seen that Open Images Dataset has the highest ratio of 8.1 for *average number of objects per image*. Moreover, its number of classes is also 600. However, among these 600 classes, OID has many classes with only 10 or 40 number of training images, which makes it not a suitable dataset for training in those cases. On the other hand, the ImageNet dataset is very popular, but it has only 200 categories for object detection, also most of them are iconic images with only one object per image, which is also not a very good factor in training neural network-based models. PascalVOC consists of only 20 classes, which are negligible in-front of millions of classes of real-world scenarios. Moreover, the number of images per class in this dataset varies highly from class to class, which makes it perfect for some classes and average for other classes. Similarly, Microsoft COCO is also a very accurate and highly popular dataset due to its performance, but 80 is still a small number. Also, these datasets have a large number of total images, but we have computed the median to find the average number of images per class in terms of training, validation, and testing. We can observe that some of the OID datasets have the least number of classes resulting in (740, 26, 77) average number of images per class. However, Microsoft COCO is best as compared to Pascal VOC and ImageNet. Also, the ratio of objects per image of Microsoft COCO is relatively high, and the image size is also the least. However, each of these datasets could be useful in terms of having different categories and can be served as a benchmark to construct base classifiers or for domain adaptation of classifiers.

Despite all of these (ImageNet, Pascal VOC, MSCOCO, and OID) imperfections, these datasets could be useful in training base classifiers. Especially performance of deep learning models by using Pascal VOC($C_{voc} = 20$) or MSCOCO ($C_{coco} = 20$) datasets is unbeatable and can perform roles of base classifiers. On the other side ImageNet ($C_{Image\_Net\_OD} = 200$) and OID ($C_{OID} = 600$) helps to construct classifiers on the need for classes which are not present in other small datasets thus they could also give an average performance which is better than having no classifier for such rarely occurred classes. If we need to detect any object which is completely unseen/unknown (i.e. $\overline{C_{voc} + C_{coco} + C_{Image\_Net\_OD} + C_{OID}}$) for any of the available datasets, then we can choose a suitable *seen/known* class ($C_{voc} + C_{coco} + C_{Image\_Net\_OD} + C_{OID}$) from available datasets, which is closest to this *unseen* class and construct a new classifier by adapting seen-class classifier as a base classifier. Lastly, if no closest class is available in popular datasets, then we can train a classifier online from scratch [156]. Although for this we may have to apply automatic data construction techniques by using search engines (Google Images, Bing Image Search API, etc.) for data collection and automatic segmentation tools for annotation.

Existing work in object detection datasets focuses on widening the vocabulary by providing a large number of annotated bounding boxes for multiple categories. Such approaches do not cover increasing the performance of machine learning models or any possible adaptation. Since datasets' construction is an independent task from object detection models, we need to bridge the large gap of performance and an extensive vocabulary.

3.3.3 Small Datasets based N-Shot learning methods

N-Shot learning is a branch of machine learning which handles the challenge of training a model with only a small amount of data. In terms of terminology, we refer to it as N-way-K-Shot-classification, where N is the number of classes and K is the number of labeled training samples from each class. N-shot learning is helpful for the domains where we have only one or two samples per class available for training. Its main variations are zero-shot learning, one-shot learning, and few-shot learning discussed below:

Zero-Shot The goal of Zero-Shot Learning is to classify a new class without any training data. In other words, a model which needs no samples to classify an image. To achieve this task, zero-shot learning [157–159] uses the metadata of images which mainly includes appearance, properties, and functionality.

One-Shot In One-Shot Learning [160, 161], we have only one sample for training. Siamese neural networks [162, 163] and matching networks are two benchmark architectures developed for one-shot learning.

Few-Shot The difference between one-shot and few-shot is that few-shot has two to five samples for training each class. Prototypical networks [164] and Meta-Transfer Learning [165] are baseline algorithms for the few-shot learning. Omniglot and Mini-ImageNet are commonly used datasets for few-shot learning. However, most of the existing methods are designed for *few-shot image classification* problems.

The Few-Shot Object Detection is less developed as compared to few-shot classification. YOLOMAML [4] is one of the open-source algorithms for the task of few-shot object detection, which utilizes two components MAML [160] algorithm and the YOLO detector [166]. Thus, it depends on classification as well as detection to perform few-shot learning. Some of the recent work [63, 167–170] in the area of few-shot learning is discussed in Chapter–8. Clearly, these algorithms are designed for tasks where few labeled samples are available for training, not for reducing the training time.

We can conclude existing N-Shot learning methods could be helpful for unseen classes, but these methods are primarily available for image classification, and object detection needs investigation for N-shot learning. In this thesis work, I handle the problem of processing unseen classes in less time within IoMT of smart cities where we can generally collect more than 2-5 samples using network sensors and require real-time processing.

3.4 Summary

This chapter analyzed the state-of-the-art of Internet of Multimedia Things (IoMT) and its comparison with the Internet of Things (IoT). Also, I provide a background of object detection with existing deep neural network-based models and fully annotated datasets. It is important to note that the construction of deep neural network object detection models and datasets are also independent tasks, so we cannot use them together to train large vocabulary models. My work aims to propose generalizable IoMT based event processing models presented in the following chapters of this thesis and demonstrate by the case study of object detection. Associated publication to this chapter is a survey [31].

[4] https://github.com/ebennequin/FewShotVision

Chapter 4

Seen/Unseen Objects based Multimedia Event Processing

4.1 Introduction

This chapter describes the proposed approach of multimedia event processing driven by existing event processing. To achieve the goal of generalizable multimedia event processing that can support unbounded vocabulary while minimizing the response-time and achieving high accuracy, I introduced different situations (i.e., scenarios) and their respective proposed models of adaptive multimedia event processing in this chapter. The scenarios are based on the category of subscription (seen or unseen), its similarity with existing concepts, and the type of data available for the training of the new classifier. I provide the rationale for our four proposed models with their hypotheses, where our main contribution lies in optimizing online testing (Model I) and online training (Model II, III, and IV) time to reduce the overall response-time for multimedia event processing. I also discuss proposed models in brief, their hypotheses, limitations, and possible solutions. However, more detail on each specific problem formulation, modules of each approach, and individual evaluations are provided in Chapters 5, 6, 7, and 8.

In this chapter, first, I introduce the approach of generalizable multimedia event processing. Next, I give detail associated with handling dynamic (seen/unseen) subscriptions in Section–4.3 to process multimedia events in smart cities. An overview of the four proposed models is discussed in Section–4.4. The discussion on limitations associated with scenarios and conclusions appear in Section–4.5. A summary of the chapter is presented in Section–4.6.

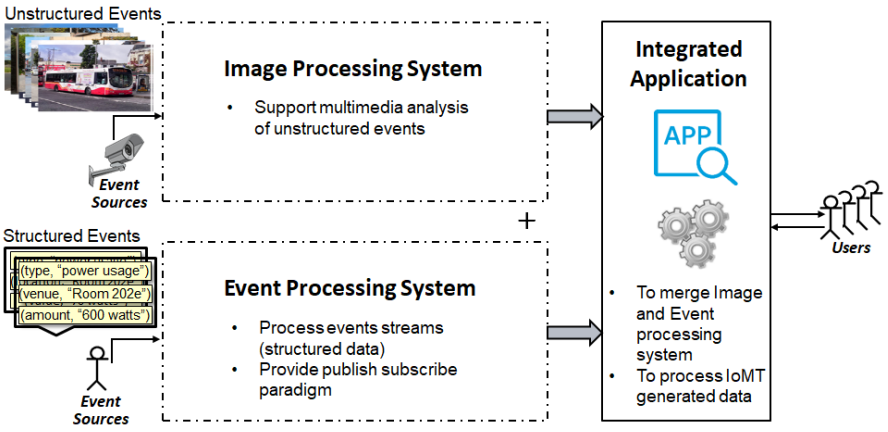


FIGURE 4.1: Current Approaches for Multimedia Event Processing

4.2 Redefining Event Processing to Multimedia Event Processing

Presently there is no generalizable system for multimedia event processing and existing image based systems are domain specific. Fig. 4.1 illustrates the current implementation of processing multimedia events within specific domains of smart cities. It consists of two key components: image processing system and event processing system to handle IoMT based events. Image based systems are responsible for analyzing images/video using domain specific feature extraction methods. On the other hand, event based systems provide the publish-subscribe paradigm to facilitate distributed interaction in large-scale applications. Thus, these two components need to be merged using another application to support user request across both systems.

In this work, I proposed a generalizable approach for handling IoMT-based events to facilitate multimedia events services irrespective of their domain. Multimedia event-processing models have been proposed within event-based services that serve as middleware between multimedia heterogeneous sensor networks and their application portal within smart cities. Fig. 4.2 demonstrates the interaction of the proposed system with wireless sensor networks for different domain-specific applications using middleware.

The proposed *multimedia event processing* is based on event processing, multimedia analysis, and deep convolutional neural networks to meet the requirements of IoMT based systems in real-time. The incorporation of event-based systems with multimedia analysis supports the processing of multimedia events streams within the publish-subscribe paradigm. Deep convolutional networks based techniques are included with multimedia analysis to facilitate the processing of IoMT generated data with high performance. A new "detect" operator has been developed to provide the requisite feature extraction to

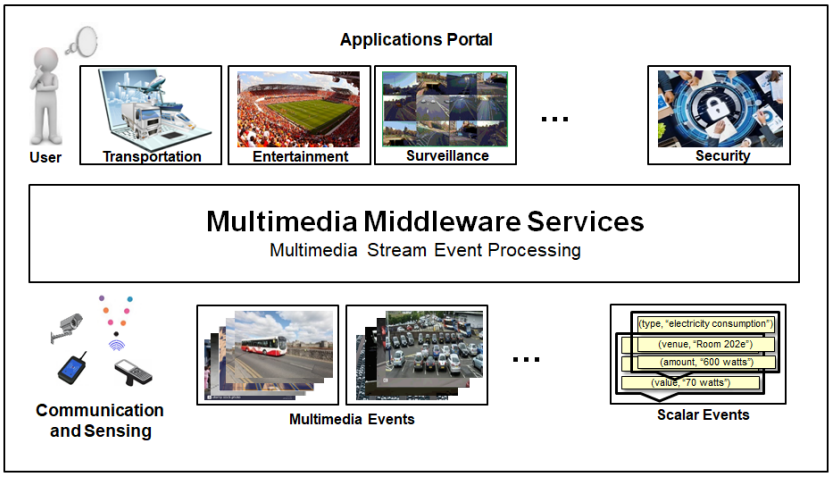


FIGURE 4.2: IoMT Aware Middleware

detect objects inside image events. The detect operator extends the event query language to support multimedia analysis. The system must design and implement an event processing engine with multimedia analysis using deep convolutional neural network-based techniques to process multimedia event streams in a publish-subscribe paradigm with high performance. Further, I used different adaptation approaches for the fast training of classifiers to answer user queries online for unseen concepts of multimedia events. We present below the concepts that we attempt to introduce in event processing for the design of multimedia event processing:

4.2.1 Defining Generalizability

I define the generalizability of an event processing engine as the ability to support different kinds of operations on multiple application events. The aim is to provide a generalizable framework that can incorporate new operations on various application events, using existing event processing query languages and feature extraction methods, irrespective of their domain and nature of events. The generalizability of multimedia event processing can be analyzed along three dimensions:

1. Nature of Events:
    - Scalar (Structured)
    - Multimedia (Unstructured)
2. Application of Events:
    - Transportation

- Entertainment
- Security
- Energy Consumption
- Temperature readings
- Other

3. Operation on Events:
   - Object Detection
   - Matching of Images or Objects
   - Localization of Objects
   - Image Classification
   - Other

In this thesis, I demonstrate my work on the multimedia event on object detection operation while considering examples of unseen concepts of multiple applications.

4.2.2 Detect Operator

The *detect* operator is a general-purpose operator that has been proposed [32] to detect objects in image events. Detect is a binary operator, which consists of two inputs: *image event* and *keyword*. Events contain the details of an image event, while keyword denotes the name of an object that the user intends to detect in an image. The return type of the operator is Boolean, either true or false, depending on the detection of an object in an image. The *detect* operator is:

boolean DETECT (Image_Event, Keyword)

Consider the situation in which a subscriber wants to know that a particular object (like car, bus, etc.) is present or absent in the current image event. Examples of such kind of queries by using the proposed *detect* operator are shown below:

Exmple 1: Query statement "*Is Bus present?*" for public transport management can be expressed as:

```
SELECT *
FROM Image_Event AS IE
WHERE DETECT(IE,'bus')
```

Exmple 2: Query statement "*Is Car absent in last 1 min?*" related to detecting the empty car parking spot where providing *time window* could be optional, can be expressed as:

```
SELECT *
FROM Image_Event.win:time(1 min) AS IE
WHERE NOT DETECT(IE,'car')
```

The "DETECT" operator is implemented using You Only Look Once [35, 149], Single Shot Detection [36], and RetinaNet [37] for the purpose of classifier based object detection.

4.2.3 Unseen Subscriptions

This dimension concerns the ability to recognize new subscriptions with the naming of objects that may not belong to the system's limited vocabulary. The lack of support for unbounded vocabularies is a bottleneck for emerging applications [171], which I am referring to as *Unseen Subscriptions*.

In order to switch from one domain (D_1) to another (D_2) for generalization, we need to transfer knowledge from the model trained on classes of D_1 (seen classes) to classes of D_2 (unseen classes) [40]. For example, if a model is trained on "bus" class for public transport management (i.e., D_1) and we want a model to detect "car" class for parking management (i.e., D_2). Then, in this case, the *bus* is a seen class, and the *car* is an unseen class. Processing/Detection of an unseen class is not the only problem in the case of public transports or parking management domains; they are a bottleneck in all domains where we want to switch from one domain to another or even switching from generalized domain (like smart cities) to a specific domain (like smart home).

4.3 Scenarios for Unseen Subscriptions of Multimedia Event Processing

We show the scenarios in Fig. 4.3, realized in this thesis for handling the problem of adaptive multimedia event processing adhering to dynamic (Seen/Unseen) subscription constraints. Suppose a user subscribes for a concept, we determine the familiarity of that concept with the multimedia event processing model by estimating "Is the Concept Unseen?". We answer the nature of the concept seen or unseen by inquiring about the availability of its classifier. If we find a classifier that can detect subscribed concept, we

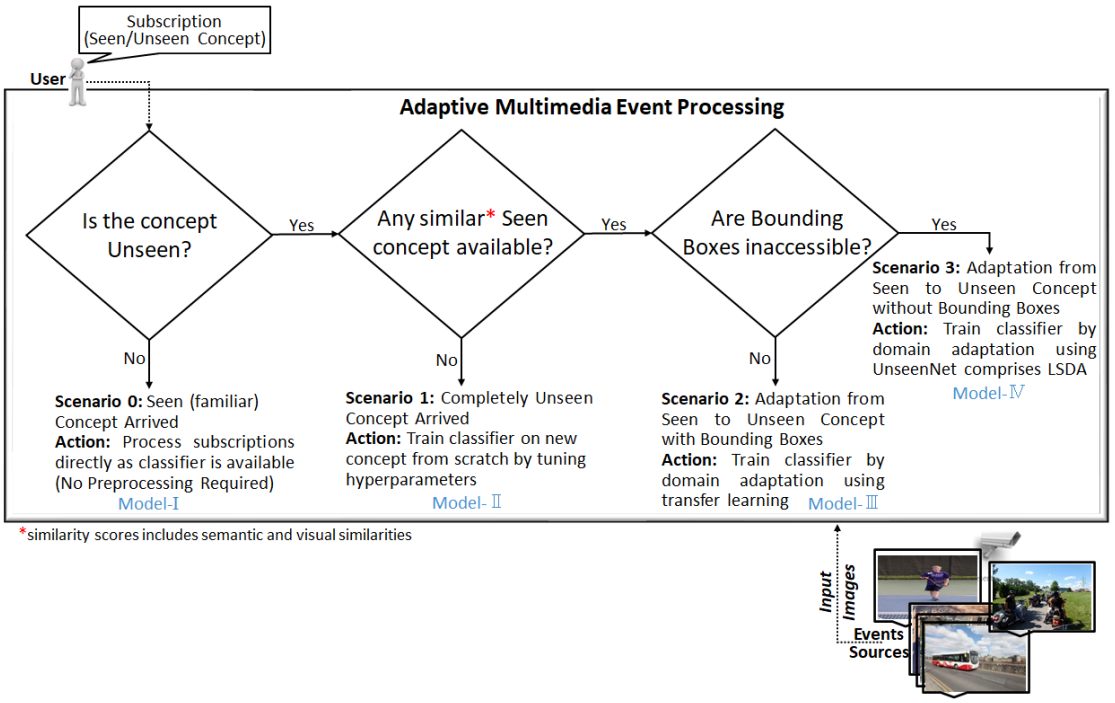


FIGURE 4.3: Scenarios for Adaptive Multimedia Event Processing adhering to Seen/Unseen Concept Problem

call the concept "Seen" and recognize it with "Scenario 0". In this case, we process the subscriptions directly using the existing classifier, without training any other model for the *seen* concept.

However, if we don't find any classifier to process the unseen concept, then we attempt to find "Any similar seen concept available?". We use the labels of existing classifiers and compute their individual similarities with the subscribed concept in this condition. In the worse case, if the concept is completely "unseen", we introduce "Scenario 1" for the handling of subscriptions that are not related to any domain and resulted in low similarity scores. Please note we use the semantic as well as visual similarities to compute the comprehensive similarity scores detailed in Sections 8.2.1 and 8.4.2.2. On the occurrence of an altogether "unseen" concept, we train the classifiers from scratch and optimize the training by hyperparameter tuning to reduce the overall response time of the multimedia event processing model.

The most likely scenario is to receive the concept which is "unseen" and have similarity with one or more "seen" concepts. Thus, we can train classifiers for such unseen domains by knowledge transfer from seen domains. Our final concern is the availability of bounding box annotations to train classifiers for subscribed concepts.

In the current scenario, we use object detection datasets to train DNN based models. Still, since the collection of bounding boxes with images is a tedious task, they have a

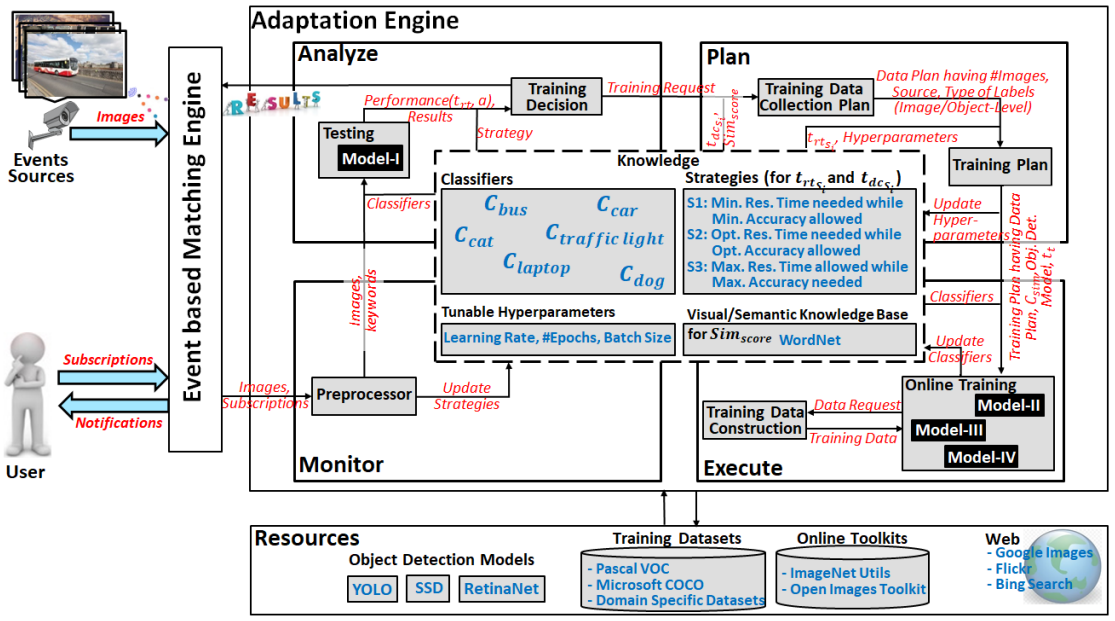


FIGURE 4.4: Adaptive Multimedia Event Processing with Proposed Models I: Domain-Specific Classifier, II: Hyper-Parameters, III: Domain Adaptation, IV: Domain Adaptation without Bounding Boxes based Multimedia Event Detection.

small number of classes or a small number of images per class (analyzed in Fig. 3.4). Most popular object detection datasets: Pascal VOC [26], MCOCO [27], OpenImages (OID) [28], and ILSVRC detection challenge [152], have only 20, 80, 600, and 200 classes, respectively. Though these datasets show promising results on the training of object detection models, they indeed fail to fulfill the data requirements of object detection models that could process numerous classes.

Since image-level labels are comparatively easy to acquire, a large number of classes can be covered easily using image classification datasets or from the web. Thus, in this work, we divide the last condition of accessibility of bounding boxes into two scenarios: Scenario 2: Adaptation from seen to an unseen concept with bounding boxes, and Scenario 3: Adaptation from seen to an unseen concept without bounding boxes. In Scenario 2, we transfer the knowledge from the seen class model to the unseen class model using transfer learning techniques while training on images having bounding box annotations.

In the last scenario of Fig. 4.3, we make use of the small number of classes based object detection datasets (having bounding boxes annotations) and the large number of classes based image classification datasets (have no bounding boxes annotations) and convert them into infinite vocabulary datasets by utilizing knowledge transfer based classifier to detector method (i.e., LSDA) [3, 30] while limiting the training time. We call our new framework of detection unseen concepts "UnseenNet" detail in Chapter–8.

4.4 Adaptive Classifier based Multimedia Event Processing Models

To handle each of the scenarios associated with the problem of multimedia event processing for the dynamic unseen concepts, I proposed different models shown in Fig. 4.4 following IBM MAPE-K architecture [172]. It consist of *Monitor*, *Analyse*, *Plan*, *Execute* phase, a shared *Knowledge Base*, and managed *resources*. The monitoring phase is responsible for interacting with the event processing engine for receiving subscriptions (in the form of keywords), image events, and any other specified requirements. The analyze phase decides on training and testing while processing images. The planning phase configures tunable parameters and generates a training plan based on the relation of unseen concepts with seen concepts. The execution phase initiates the training of the classifier (if necessary). The knowledge base consists of baseline classifiers, strategies, hyperparameters, visual/semantic similarity of seen and unseen concepts. The last layer is to place the existing online/offline training datasets and Deep Neural Network (DNN) based models. We take "object detection" as a case study to demonstrate the efficiency and limitations of our models for tasks in smart cities.

In Model I, I use the existing architecture of the event processing engine and introduced the ability to process multimedia events while optimizing the *testing* time. The optimization is based on the inclusion of "classifier division and selection approach," which enables the proposed feature extraction model to choose a suitable classifier based on subscription constraints, ultimately *optimizing* the *testing*. Next, I analyze the problem of adaptation to handle completely or partially unseen concepts. The Model II uses hyperparameter tuning to achieve the reduced response time for the training of completely unseen concepts. In this work, I also detail the self-adaptation architecture and show the *optimization* of *training*. Finally this thesis focus on problem specific to domain adaptation for the remaining scenarios (Fig. 4.3). Here, I optimize the training further in model III and IV by knowledge transfer for both cases of with and without bounding boxes. More details of proposed models with their hypotheses are discussed in this section.

In all research questions, I attempt to minimize the response time and maximize the accuracy. Thus, we can observe a trade-off between response time and accuracy due to the different characteristics of object detection models, as accurate object detection models are slower presently. This trade-off cannot be categorized as "Pareto efficiency" due to two significant reasons. First, from the perspective of testing time, if object detection model has more inference time and high accuracy, then it is not necessary that its accuracy cannot be increased without increasing inference time, as many other

factors like network architecture, type of data, type of resources, etc. can affect the performance. For example, RetinaNet [37] is faster and more accurate than the Faster R-CNN [148] object detection model because of its better architecture. Secondly, from the perspective of training time, accuracy may or may not increase with training time, depending on the type of transfer learning techniques used for training, kind of training data, type of source model available, etc. Moreover, if accuracy increases, it may stop increasing after some time regardless of increasing the training time, as it is always between 0 and 1. Thus, we cannot guarantee that accuracy cannot be increased without hurting response time; it may or may not improve with efficient neural network-based models.

4.4.1 Domain-Specific Classifier based Multimedia Event Detection Model

The first model tackles the requirement of high accuracy and low response time by having a contribution in optimizing testing (inference) time. It can be observed that existing event processing systems focus on structured (scalar) events for processing of subscriptions of a user and have no provision of handling multimedia data. There is also a limited provision of multimedia query languages for events. At the same time, image processing systems are domain-specific, thus not generalizable. I frame this problem as the first research question "*RQ1: How can we answer multimedia event based queries online consisting of seen concepts of any domain while achieving high accuracy and minimizing the response time?*".

I formulate the hypothesis of this model as: *Domain-Specific classifier based multimedia event processing assumes that if we construct N-Class classifiers for different domains, and we use subscription constraints to choose closely related classifier for the processing of multimedia events; the performance will be enhanced in terms of accuracy and response time, and will also add the ability to generalize for multiple domains.*

The first model focuses on the processing of seen concepts (Scenario 0) with high performance. However, the proposed model requires trained classifiers to process multimedia events. Conventionally, these classifiers are trained on general-purpose datasets consisting of a large number of classes related or not related to domains. In this hypothesis, I assume that if we use N-Class classifiers for individual domains instead of using a single classifier consisting of classes of all domains, we could enhance performance. It will also make our model flexible for adding more domains on need. Thus, an optimization based on the inclusion of the "classifier division and selection approach" is proposed, enabling the proposed multimedia event processing model to choose a suitable classifier based on

subscription constraints. The detection model extract objects using only specific classifiers related to the prescribed attributes (keywords). For instance, "car" classifier (single class classifier) will be selected by the "YOLO" model for the detection of a *car*. Due to the same reason, Model-I is named as *Domain-Specific Classifier* based multimedia event detection model.

This model has been presented in the journal of IEEE Access "Towards a generalized approach for deep neural network based event processing for the internet of multimedia things" [32] and will be discussed further in Chapter–5.

4.4.2 Hyper-Parameters based Adaptive Multimedia Event Detection Model

In the previous model, we optimized the generalizable multimedia event processing for seen concepts and realized the challenge of online training for unseen concepts. Model II focuses on adaptive multimedia event processing, where we can process concepts that are completely unseen (Scenario 1) and tune hyperparameters to optimize the training. This problem is formulated as second research question "*RQ2: How can we answer multimedia event based queries online consisting of completely unseen subscriptions (unbounded vocabulary), using an adaptive classifier construction approach with tuning of hyper-parameters while achieving high accuracy and minimizing the response time?*".

To address the problem, I incorporated multimedia event processing with an *adaptation* engine and *online classifier learning* based object detection methods to meet the requirements of dynamic seen/unseen concepts. Since the choice of hyperparameter values greatly affects the performance of resulting classifiers, we leverage hyperparameter tuning based techniques, including the configuration of learning-rate, batch-size, and the number of epochs for minimizing the response time.

Hypothesis for Model-II can be framed as: *If tuning of hyperparameters based technique is useful in machine learning models to speed-up the training, decrease the computation cost, and increase the accuracy; then performance will get enhanced for low response-time also even on training from scratch for unseen subscriptions on tuning hyperparameters for the online construction of classifiers.*

This model has been presented in the journal titled "Investigating response time and accuracy in online classifier learning for multimedia publish-subscribe systems [33]" of Multimedia tools and applications (MTAP) Springer and will be discussed further in Chapter–6.

4.4.3 Domain Adaptation based Multimedia Event Detection Model

Consider an option of online training of classifiers on request of any unseen (new) subscription, which will require either switching (transforming) from one classifier to another (like bus → car) or the construction of a completely new classifier (like ball). The previous Model-II covers the latter option. In present Model-III, we analyze the former option of transferring knowledge from one classifier to another, representing Scenario 2 in Fig. 4.3. For partial unseen concepts, there is a need to investigate the online construction of classifiers that can allow adaptation among domains for seen/unseen concepts considering overall response time (including training time) and accuracy.

This problem is devised in research question "*RQ3(a): How can we answer multimedia event based queries online consisting of unseen subscriptions (unbounded vocabulary), using domain adaptive classifier construction approach with knowledge transfer from seen subscriptions (bounded vocabulary) while achieving high accuracy and minimizing the response time?*".

In this work, I incorporated transfer learning based techniques in the proposed adaptive multimedia event processing engine. We investigated the two knowledge transfer methods: (1) fine-tuning pre-trained models and (2) freezing backbone layers of similar classifier while training only top dense layers of object detection models.

The research hypothesis for Model-III is: *Domain adaptation based Multimedia Event Detection model relies on the fact that if transferring of knowledge from one domain to another (say $A \rightarrow B$) can improve the performance as compared to fine-tuning of pre-trained models (like $C_{P_{ImageNet} \rightarrow B}$) or training of classifier from scratch (C_B); then there will always be a decrease in response-time with increase in accuracy of constructed classifier ($C_{A \rightarrow B}$) than the classifier trained from pretrained model (like $C_{P_{ImageNet} \rightarrow B}$) or training from scratch (C_B).*

This model has been presented as the short paper titled "Reducing response time for multimedia event processing using domain adaptation [34]" at ACM ICMR 2020, journal titled "Detecting Seen/Unseen Concepts while Reducing Response Time using Domain Transfer in Multimedia Event Processing" of IEEE Access (Under Submission), and will be discussed further in Chapter–7.

4.4.4 Domain Adaptation based Multimedia Event Detection Model without Bounding Boxes

All previous models assume we have images with bounding box annotations available to train neural network-based models. However, this is not a realistic scenario as most

object detection datasets have limited (bounded vocabulary). For instance, Pascal VOC has 20, Microsoft COCO has 80, and OpenImages have 600 classes. Thus, it is impossible to arrange object-level annotations (i.e., annotated bounding boxes) for online training of models for all unseen concepts. The final objective of this thesis is to train detectors for any possible unseen concept (i.e., an infinite number of classes) without bounding box annotations within a limited amount of time.

I formulate the final specific problem in research question as "*RQ3(b): How can we answer multimedia event-based queries online consisting of unseen subscriptions (unbounded vocabulary), using task as well as visual domain adaptive classifier construction approach with knowledge transfer from seen subscriptions (bounded vocabulary) while eliminating the requirement of bounding box annotations availability, achieving high accuracy, and minimizing the response time?*".

I proposed an "UnseenNet" detector (Model-IV) to handle the problem of weakly supervised learning while optimizing training without bounding boxes. Proposed model is based on making use of existing object detection datasets of bounded vocabulary (consists of seen concepts) to construct detectors for unseen concepts (i.e., unbounded vocabulary) by using the differences between a weak detector (trained on image classification dataset, i.e., image-level labels) and a strong detector (trained on object detection datasets, i.e., object-level labels). The UnseenNet also uses the concept of Large Scale Detection through Adaptation (LSDA) based approach to eliminate the need for bounding boxes, and similarity between classes enhances the accuracy within less training time on knowledge transfer.

These assumptions are expressed in the research hypothesis for Model-IV as: *If an adaptation of classifier into detector eliminates the need of bounding boxes as well as transferring of knowledge from one domain to another speed-up the training; and a detector gets constructed from classifier with the help of transfer of knowledge from visually/semantically similar classifier; then that detector will take less time to train for unseen classes and eliminate the requirement of bounding boxes.*

This model has been presented in the paper titled "UnseenNet: LSDA-based Fast Training Detector for Unseen Concepts with No Bounding Boxes", currently under submission at IEEE TPAMI, and will be discussed in detail in Chapter–8.

4.4.5 Deployment of Multimedia Event Processing Models

As multimedia event processing models are designed to process IoMT based data, they can provide event-based services while serving as middleware between sensor networks

and their application portals within smart cities (see Fig. 4.2). The enhanced event processing for IoMT events at the middleware enables the analysis of multimedia (unstructured) data and scalar (structured) data. Proposed models at IoMT-aware middleware can consume events (scalar/multimedia) generated from sensors (like camera, RFID tags, temperature sensors, etc.), process them using multimedia event processing engine, and react to users accessing multiple types of applications. Model-I is applicable for seen classes only, reducing the response time to 0.009 seconds (114fps). Moreover, it uses N-class classifiers; thus, it will use a single classifier for processing N classes which will ultimately require low memory. Therefore, Model-I is applicable for resource-constraints devices work at the edge of the network. Model-II will mainly require training from scratch; thus, it will need GPU for its first response to the unseen subscription. However, once the model gets trained, testing could be deployed to edge devices. Model-III utilizes domain adaptation to reduce further the response time, which requires finetuning the model for transfer learning. Such finetuning can be done on the CPU but may take a longer training time to respond to unseen subscriptions. However, once the classifier gets ready, I believe Model-III can also be used (like Model-II) on resource-constraint devices. However, Model-IV (UnseenNet) is the lightest among all models and takes only 5 min for training. Thus, I assume it will be best among all other models (I, II, and III) to get deployed on the network's edges and analyze the performance. In conclusion, proposed models are designed for multimedia-aware middleware cloud to support multimedia event services and can be further improved for edge-based servers.

4.5 Discussion

Other than our four models that cover all scenarios we discuss here few elements that we considered in our approaches but not from the scope of contribution. We describe them briefly in this section with their use, assumptions, and our possible solutions.

Use of Online Toolkits: On request of an unseen class, we suggest that the system collect images from the Web using Google Images[1], Flickr[2], or Bing Image[3] search. However, in experiments of downloading data from the Web to train models without bounding boxes, we find that such search API also leads to few not-reliable images. Though these useless images are less in number, we nevertheless recommend researchers first consider the option of online toolkit ImageNet-Utils[4] and OIDv4_ToolKit[5]. In

[1]https://github.com/hardikvasa/google-images-download
[2]https://www.flickr.com/services/api/
[3]https://pypi.org/project/bing-image-downloader/
[4]https://github.com/tzutalin/ImageNet_Utils
[5]https://github.com/EscVM/OIDv4_ToolKit

this case, the ImageNet toolkit will allow downloading images without bounding boxes for more than 1000 categories using their WordNet Ids. Moreover, the OID toolkit will give access to more than 600 categories with bounding boxes. It is also worth considering ImageNet Downloader[6], as it is described in a study[7] that mostly Flickr URLs of ImageNet classification dataset are useful for successful downloading of data compared to other URLs. In our experiments of collection of 100 unseen classes, we also found that many WordNet IDs do not allow users to download images of ImageNet, as they are still under construction.

Amount of Training Data: It is a very common question of machine learning problems: "How much data we need to train". However, the answer is different for everyone. It is highly dependent on the problem and the learning algorithm. In our case of object detection, we are highly dependent on the number of images present in object detection datasets. These numbers of images are very different for each class (detail in Fig. 3.4), varying from 40000 for the "person" class and 3 for the "hair drier" class. Due to this reason, we found in our implementation that our models are getting biased towards the "person" class and other classes that have a drastically higher number of training images. However, the number of such classes is less than 5; all other classes have less than 5000 images. Thus, in our training datasets of seen and unseen classes, we consider only up to 5000 images for each class.

Another critical point to consider is that downloading time could also decrease/increase the response time. However, fewer images could need more iterations to train the model, whereas more images need fewer iterations. One iteration is equal to the $number\ of\ images/batch\_size$. Thus more iterations will need more training time. Investigating one-shot learning [173] is a reasonable future direction before reducing the number of training images.

N-Class Classifiers: Our optimization on testing time is based on using n-class classifiers belonging to a particular domain related to subscriptions. For instance, in the case of n=1, the "car" classifier (single class classifier) will be selected by model for the detection of a subscription *car*. The basic idea is to use only the available classifier, which is closely related to the concepts within a subscription, which can vary from single to n-class classifiers having n ranges 1 to ∞. We show that the performance will decrease with an increase in the number of classes per classifier. Thus, it is beneficial to choose the optimal value of "n" and consider only the related classes for constructing a classifier for optimization. We prove that choosing lower values of "N" related to the application domain can improve the throughput without influencing its accuracy.

[6]https://github.com/mf1024/ImageNet-datasets-downloader
[7]https://towardsdatascience.com/how-to-scrape-the-imagenet-f309e02de1f4

Space Complexity: We construct different detectors for each class. Suppose our model receives a new subscription, "donkey". When we fine-tune/adapt the whole network on image-level or object-level labels of "donkey", the network will be biased towards detecting class "donkey" while other seen classes will become less accurate. Next, if our model receives another new subscription like "monkey". After fine-tuning/adapting the network on data of "monkey", the network will be biased towards detecting class "monkey". Simultaneously, other classes (including the "donkey" class) will become less accurate. Consequently, the resulting performance for two subscribers subscribing "donkey" and "monkey" will start depending on each other, which is unintuitive and should not be the case. Due to this reason, we keep detectors of different subscriptions separated and construct every time a specialized detector for the particular subscription.

An important question emanates from this approach is "Isn't the space complexity too high if we keep constructing different detectors?". The answer is probably "no". Assuming the average size of the detector is 100MB, we get 1 million unseen concepts (which is already highly unlikely). Then total space detectors will occupy $=100\text{MB}\times10^6=100\text{TB}$, which is not too large space in the era of having 1TB hard disk on personal laptops when we can handle millions of classes.

Same class and different names problem: It is essential to specify that image classification datasets (like ImageNet) and object detection datasets (like MCOCO, OID, Pascal VOC) use different names for the same classes. So, we use the vocabulary of WordNet to give a single name to each class and provide mappings of these datasets and WordNet along with our proposed models. Thus, our all seen and possible unseen classes belong to a single WordNet vocabulary.

4.6 Summary

This chapter redefined event processing to multimedia event processing and introduced the concept of generalizability, detect operator, and unseen subscriptions. Then, I analyzed the scenarios of handling unseen subscriptions for multimedia event processing. Each of the scenarios gives the rationale of our four main models: (1) domain-specific classifier processing, (2) hyperparameter tuning, (3) domain adaptive processing with, and (4) without bounding boxes, proposed in this thesis. Proposed models are based on three conditions: the occurrence of unseen concepts, the presence of similar seen concepts, and accessibility of bounding box annotations for the online training of models. Our first model is responsible for the generalizable multimedia event processing engine (Model I) and proof of optimization on using domain-specific classifiers. The adaptation approach (Model II) is useful for handling any completely unseen subscription where

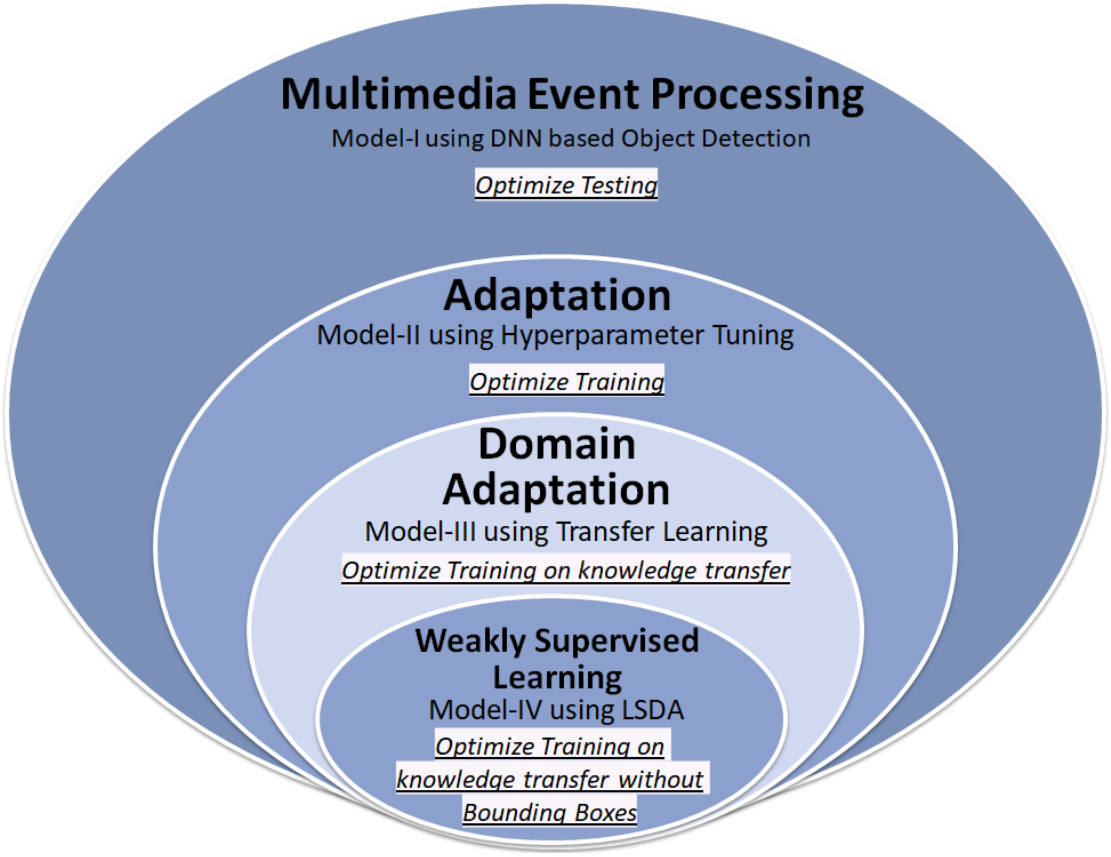


FIGURE 4.5: Summarizing Proposed Techniques for Adaptive Multimedia Event Processing

no similar seen concept is available. I optimize the online training, in this case, using the tuning of hyperparameters. The most crucial reason for Model III and IV is to allow the adaptation among domains to construct detector for *unseen* class from the detector of *seen* class, to speed up the training (i.e., decreasing the response time) while increasing the accuracy. The availability of annotated bounding boxes with images is not likely to happen scenario in most cases of unseen concepts. Thus we proposed two different domain adaptation approaches to cover both cases. I also discussed their hypotheses, problems, limitations, and assumptions. I summarize the proposed techniques associated with the problem of multimedia event processing for the dynamic unseen concepts in Fig. 4.5. The proposed models with specific techniques, research questions, and evaluations are discussed in more detail in Chapters 5, 6, 7, and 8.

Chapter 5

Domain-Specific Classifier based Multimedia Event Detection

5.1 Introduction

I analyzed the problem of adaptive multimedia event processing in Chapter–4. Among different presented scenarios, this chapter analyze the foremost basic scenario related to the Research Question 1: "*How can we answer multimedia event based queries online consisting of seen concepts of any domain while achieving high accuracy and minimizing the response time?*" in Section–5.2. This chapter tests the Hypothesis-I: "*Domain-Specific classifier based multimedia event processing assumes that if we construct N-Class classifiers for different domains, and we use subscription constraints to choose closely related classifier for the processing of multimedia events; the performance will get enhanced in terms of accuracy and response time, and will also add the ability to generalize for multiple domains.*" formulated in Section–1.6.

Before the proposed approach, I discuss the background (in Section–5.3) and divide the related work into three major categories: *Event-based Approaches for IoT*, *Application-Specific Approaches for IoMT*, and *Multimedia Query Languages*. I analyze event processing approaches that are efficient for scalar events and do not consider multimedia events. On the other hand, accurate multimedia event processing approaches are domain-specific with limited maintainability. To tests the hypothesis, I proposed a generalizable multimedia event processing engine describe in Section–5.4 and 5.5. The proposed Multimedia Event Processing Engine (MEPE) along with an optimization technique, consists of neural network based feature extraction operators, and extends event query language to support multimedia analysis within event-based systems. Within MEPE the user can define subscriptions using the proposed *detect* operator based on

object detection. The subscription is used by the object detection model to choose the relevant classifier which is needed to identify the prescribed attribute, which we describe in the "*classifier division and selection*" approach. I proposed an optimization model which uses subscriptions at two different stages: (1) analyzing the query, and (2) optimize the processing of a neural network based matcher based on subscriptions constraints. The resulting model (Section–5.6) is proficient in processing multimedia event streams belongs to multiple applications while achieving high throughput and comparable accuracy which also confirms its generalizability within smart cities infrastructure.

5.2 Problem Overview

5.2.1 Preliminaries

- Publish/Subscribe is a message-oriented interaction paradigm in which publishers send messages to the middleware, and the consumers express their particular interest in receiving some useful information [174].
- Middleware provides general-purpose services between distributed (multiple) platforms and domain-specific applications. The main goal of middleware is to enable the interaction and communication between distributed components, hiding from application developers the complexity of the underlying hardware and network platforms and freeing them from explicit manipulation of protocols and infrastructure services [48].
- Event-based middlewares consist of rich literature for structured event processing and managed to bring an uprising change in the communication models of distributed systems [175]. Event processing systems introduced to process event streams within publish/subscribe paradigm. They are based on tracking and analyzing (processing) streams of information about things that happen (events) and deriving a conclusion from them [8]. Multiple entities used in literature to define the term *event processing* are described as follows:
  - ***Events***: An event is anything that happens and is a significant observable occurrence. It is also called a message containing the information, which is in the structured form conventionally. These types of events are published by publishers, detected by sensors, processed by event-based matcher, and finally consumed by consumers.
  - ***Subscriptions***: Subscription is a registration and association of an event action to indicate that a particular event is of interest to the user. Usually, it

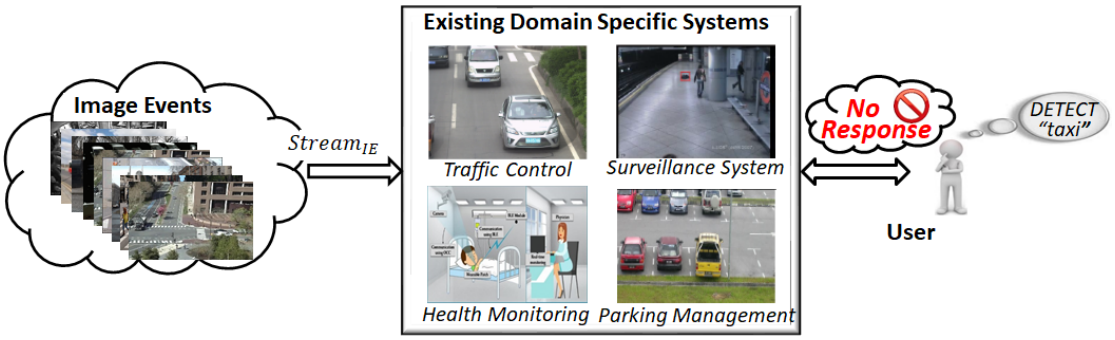


FIGURE 5.1: Multimedia Event Processing in Existing Domain Specific Applications

is an expression of subscribers to match an event if all primitive constraints are satisfied.

- ***Matcher***: The matcher is responsible for analyzing events and communicating them according to applicable subscriptions. It is also known with similar terms like the broker, servers, routers, etc. In an event paradigm, a matcher will detect single events or patterns of events depending on the matcher's complexity.

- Internet of Multimedia Things (IoMT) can be defined as an IoT-based paradigm which allows objects to connect and exchange structured as well as unstructured data with one another to facilitate multimedia-based services and applications [5].

5.2.2 Motivational Scenarios

Consider event detection scenarios (shown in Fig. 5.1) of smart cities, where we have multiple applications like traffic control, health monitoring, parking management, or any other surveillance systems. Suppose a user subscribes for the detection of "taxi", then none of the applications will be able to process the query even when many of them can recognize the "car". Presently, these multimedia-based communication technologies are domain-specific, and research on IoT mainly focuses on handling big data challenges, excluding multimedia, leaving a gap between the advancement of IoT and multimedia-based technologies. Thus we need a classifier-based approach that can answer a large number of concepts for any domain in low response time. This represents the first baseline scenario of detecting seen concepts for analyzing multimedia events, as shown in Fig. 5.2. In this scenario, we assume that if the concept is seen, we can process it directly by having trained classifiers available offline according to their domains.

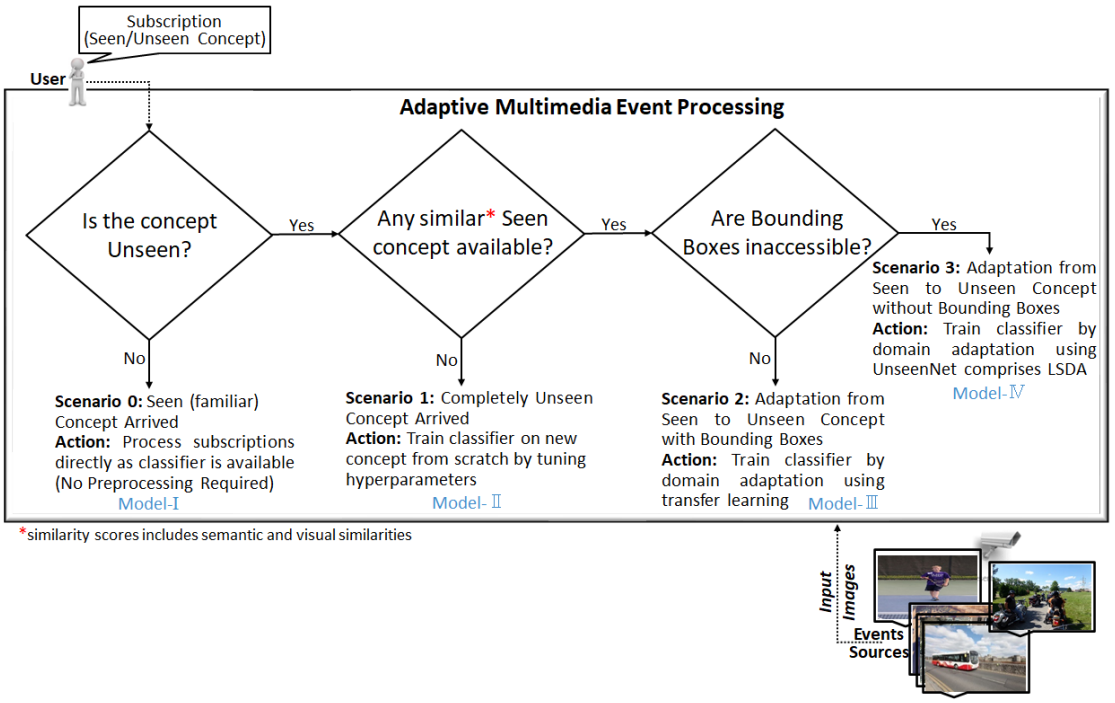


FIGURE 5.2: Scenarios for Multimedia Event Processing adhering to Seen/Unseen Concept Problem

5.2.3 Problem Statement

How to design a generalized event processing system that can consume Internet of Multimedia Things (IoMT) generated data as a native event type, support multimedia analytics, and react to users situations of interest with high performance?

5.3 Background and Related Work

I analyze the state-of-the-art event-based middleware methods and their suitability for the IoT as well as IoMT based events in this Section. I also review a wide range of application-specific middlewares for multimedia-based applications that could be suitable for IoMT. Moreover, incorporating multimedia query languages to represent and define multimedia events in real-time applications is useful for processing multimedia events [176].

5.3.1 Event-based Approaches for IoT

In event-based middlewares, all components, applications, and participants communicate through events. Message-oriented middleware (MOM) is one of the types of middleware,

where communication relies on messages. It follows the publish/subscribe paradigm. Event processing systems process the subscription of a user based on standard languages in response to events. Most of the existing event-based middlewares (discussed in this Section) only focused on scalar (structured) events for the processing of subscriptions of a user and with limited provision of handling multimedia (unstructured) events. The most popular event-based approaches (SIENA [177], CEA [178]) rely on producer-consumer paradigm while utilising mediator for providing services and works for supporting application-specific structures.

SIENA [177, 179] is also an event notification service designed for event-based systems with the aim of high expressiveness and scalability. It also works for application-specific attributes with best-effort real-time performance. The Cambridge Event Architecture (CEA) [178] also provides a middleware platform that allows producers and consumers to interact using event-based operations. It extends the middleware by providing a flexible and scalable approach for distributed applications. Other than supporting application-specific structures, it also fulfills the requirement of timely response to asynchronous events, which is crucial to smart cities-based applications. SECO [180] model is based on building a fully distributed version of the ECO [181] event model including event filtering capabilities. Thus its implementation is named SECO for Scalable ECO. The ECO model simply consists of three central concepts (events, constraints, and objects), and its application programmer interface contains only three operations (Subscribe, Raise, and Unsubscribe). SECO provides the best-effort quality of service for supporting different real-time application domains.

An object-oriented infrastructure called Java Event-based Distributed Infrastructure (JEDI) [182] supports the development and operation of event-based systems. It also follows the publish/subscribe paradigm to implement OPSS (ORCHESTRA Process Support System) workflow management system [183–185]. Hermes [186] is also an event-based distributed middleware that follows a type and attributes based publish/subscribe model. It introduces the notion of an overlay routing network, where producers and consumers connect to the broker network, and individual brokers subsequently route events through the overlay network. It also attempts to bridge the semantic gap between events and programming language types for the high expressibility of the model. STEAM (Scalable Timed Events And Mobility) [187–190] is an event-based middleware service that specially designed for wireless local area networks utilizing the ad hoc network model. STEAM, allows us to define delivery deadlines and assign them to specific events then dispatcher exploits these deadlines for the timely delivery of events to the subscribers. Moreover, it also allows applications to associate specific attributes either to an event type or to a specific event instance.

The Toronto Publish/Subscribe System (TOPSS) [183, 184] gives an overview of the publish/subscribe paradigm, analyzed mobile application requirements, and also implements content-based publish/subscribe paradigm with high throughput. Specifically, it addresses two key requirements (Scalability and Ability to Support Changes) to publish/subscribe middleware raised by emerging mobile applications. Semantic Toronto Publish/Subscribe System (S-ToPSS) [185] is designed to address the problem of semantic matching. Although this work demonstrates the matching of unrelated objects seamlessly, details about the workload and evaluation criteria are still part of future work. An Ontology-Based Publish/Subscribe System (OPS) [191] attempts to improve the expressiveness of the publish/subscribe system without sacrificing efficiency. It describes the data model, subscription language, matching algorithm and maintain high matching efficiency. However, it assumes only one ontology with a relatively small number of classes and properties and provides some loose coupling in developing the rules. A Publish/ Subscribe System Supporting Approximate Matching (A-TOPSS) [192–194] addresses the requirement of providing a publish/subscribe data model with an approximate matching scheme that allows the expression and processing of uncertainties for both subscriptions and publications. It utilizes Fuzzy set theory and possibility theory to represent uncertainties in predicates and publications. The language model of A-TOPSS is flexible and powerful in that it allows subscriptions and publications to be either crisp or approximate. Moreover, its effectiveness and efficiency are also relatively high.

GREEN (Generic & Re-configurable EvEnt Notification service) [195] is a highly configurable and reconfigurable publish-subscribe middleware to support pervasive computing applications. It is a highly dynamic middleware that addresses such requirements of configurability and reconfigurability requirements of such heterogeneous and changing environments. Although there are relative performance trade-offs for different configurations, the model still provides high performance irrespective of its flexibility. EMMA (Epidemic Messaging Middleware for Ad hoc networks) [196] is an adaptation of Java Message Service (JMS) for mobile ad hoc environments. It has numerous practical application domains in allowing inter-community communication in extreme scenarios of partially connected mobile ad hoc networks. Other than the autonomous behavior of EMMA, it also provides high performance in terms of delivery and latency. Mires [197] incorporates characteristics of Message-Oriented Middleware (MOM) by allowing applications to communicate using the publish-subscribe paradigm. It encapsulates the network-level protocols (routing and topology control protocols) and provides a high-level API that facilitates the development of applications over WSNs. It does not provide real-time services and also does not support dynamic behavior. SensorBus [198] is also a MOM-based model for WSNs that follows the publish-subscribe paradigm and allows

free exchange of the communication mechanism among sensor nodes. To address the requirements of a larger number of applications, SensorBus adds the capability of using more than one communication mechanism as each communication mechanism provided by a determined routing protocol in WSNs is application-specific. However, it also neither provides real-time services nor dynamic behavior support.

MiSense [199, 200] is a energy-efficient middleware architecture. It utilizes a low-power communication model and an energy-efficient resource allocation technique to achieve high throughput for WSN. Rebeca [201–204] event-based middleware is used to implement the notion of scopes into the event-based services. Visibility control within distributed event-based applications referred to as *scoping*. Although heterogeneity is handled manually in this model, efficiency is high, and also it allows loose semantic to support large vocabulary. PSWare [205] is a publish/subscribe based middleware developed to support both primitive and composite events in WSN. It also contributes to the development of a runtime environment on sensor nodes. Moreover, its event detection language can achieve high expressiveness and availability.

TinyDDS [206] is an interoperable and a pluggable publish/subscribe framework designed for event-based middleware. It allows WSN applications to have control over application-level and middleware-level non-functional properties. TinyDDS provides two types of interoperability: programming language interoperability and protocol interoperability. It is lightweight and efficient but does not address the heterogeneity and adaptation based requirements. PRISMA [207] is a resource-oriented middleware for Wireless Sensor Networks (WSN) that also follows the Publish-Subscribe paradigm. Event-based model PRISMA utilizes REpresentational State Transfer (REST) [208] for defining lightweight communication between applications. The main goals of Prisma include: (i) programming abstraction, (ii) topology control, asynchronous communication, and resource discovery services, (iii) runtime support, and (iv) QoS mechanisms. However, preliminary evaluations of Prisma do not validate their real-time or dynamic behavior. Approximate semantic matching [135, 209, 210] is among one of the recent methods which examines the requirement of event semantic decoupling and investigated the approximate semantic event matching with its consequences. It introduced a semantic event matcher while utilising thesauri-based and distributional semantics-based similarity and relatedness measures. Although such methods are focused on heterogeneity of events, such event-based systems only focus on structured events for the processing of subscriptions of a user and with no provision of handling the feature extraction requirements of multimedia events.

High-speed nature of event streams with high bandwidth of multimedia data also requires the incorporation of the optimisation techniques in existing event-based systems.

However, most common optimisation techniques [211–213] in these event processing systems are generally based on predicate indexing and network algorithms of matching subscriptions. Predicate indexing algorithms [214–217] are structured in two phases. The first phase is used to decompose subscriptions into elementary constraints and determine which constraints are satisfied by the notification. In the second phase, the results of the first phase are used to determine the filters in which all constraints match the event. However, the indexing in these approaches are based on the schema of events, and multimedia events are schema-less. Testing network algorithms [218–221] are based on a pre-processing of the set of subscriptions that builds a data structure composed by nodes representing the constraints in each filter. The structure is traversed in a second phase of the algorithm, by matching the event against each constraint. An event matches a filter when the data structure is entirely traversed by it. Predicate based grouping in these algorithms is based on attribute values and thus can fail to support multimedia event processing.

It can be observed that all of these existing publish-subscribe based event processing systems are only focused on structured (scalar) events for the processing of subscriptions of a user, with no provision of handling and optimising events consisting of multimedia data.

5.3.2 Application-Specific Approaches for IoMT

Multimedia processing (mostly image/videos) is one of the most common types of events within the applications of smart cities [144]. Since the event-based analysis of multimedia content is among the keen research areas, numerous solutions have been proposed for different application domains. However, multimedia processing events refer to "An event is the representation of a change of state in a multimedia item planned and attended [50]". Events in multimedia content are also described as "real world happening planned and attended by people". In a broader perspective, event-based analysis in the case of multimedia realized as a monitoring application. Multimedia event processing systems generally provide high performance but have domain-specific characteristics and cannot be adapted to multiple domains. Thus, merging of event-based middlewares using IoT [8, 48] with image processing systems each time with the change of application domain becomes an essential step in their deployment of smart cities. This requirement also limits the performance and requires a high setup cost.

Multiple applications are designed for different roles like traffic management, parking, surveillance, health monitoring, and various supervision activities in smart cities. These

numerous applications are highly efficient in processing multimedia (unstructured) systems events and provides real-time performance with high accuracy. For instance, traffic management based applications is highly efficient in detecting and analyzing traffic events. A traffic recognition system and traffic congestion prediction system presented in paper [10, 222], mainly constructed to automatically inform the driver about the traffic sign and for reporting predicted traffic congestions. However, from the perspective of the requirements of the problem "adaptive multimedia event processing", no support for large vocabulary limits their user interface and suitability for maintainability. A real-time traffic sign recognition scheme [222] is proposed to assist driving using smartphones. The proposed model includes five different stages: (1) video/frames capturing using a smart-phone, Notebook, and other computer devices, (2) then preprocessing stage improves the image quality and perform normalisation operations, (3) traffic sign detection step monitors frames to detect the region of traffic signs if they exist, (4) extracts the detected sign, and (5) finally a model recognises the character/icon. Their experiments proved that the model could achieve accuracy up to 98% while having a recognition speed of 0.085 seconds per frame. However, this model is applicable only for *traffic-signs* detection, and even those signs assumed to have either rectangular or circular shape. Similarly, another research [223] focused on the problem of traffic light switching according to traffic congestion on the road. This system consists of 4 video cameras on the traffic junctions; then it takes one image of the empty lane as a baseline to compute the density of vehicles on the road. Then it keeps monitoring the density of vehicles present every second, for all the lanes where light is red. Then the time for the green light signal is calculated using the number of vehicles that can pass in one second using the records of density. Results for traffic light switching show that the model can improve the time for passing the vehicles up to 35% approximately. Another work [224] intends to provide the Internet of Vehicles (IoV) based traffic management solution. Proposed IoV focused on communications of four types: communication between the vehicles and the vehicle owners, communication between vehicles, communication between vehicles and a centralised server and communication between the server and third parties (emergency response, pollution control, police patrol). Advantages of IoV include *traffic control*, *human proximity detection*, *theft avoidance*, *accident avoidance*, *emergency response*, and *vehicles-autonomous*. Its identified drawback is related to *security* and *failure of networks*.

Based on real time video analysis, a real time event detector [225] is constructed for each action of interest by learning a cascade off filters based on volumetric features that efficiently scan video sequences in space and time. The presented system follows the model based approach for event detection, for constructing a framework to analyze videos efficiently. Another technique [226] targets video stream for the analysis of moving objects

in the scene. It also provides a configuration for event detection and behavior analysis of video-surveillance streams. Due to lack of provision of publish-subscribe paradigm, the merging of event based systems with image processing systems is an essential requirement. However, the need of integration of event based model for the processing of videos, limits the performance of the system and also requires high setup cost. Such video event monitoring systems [10, 222, 223] are developed for real-time traffic management in smart cities. These systems attempt to process multimedia (unstructured) events with high efficiency, but most are domain-specific and cannot be generalized for multiple applications.

Similarly, an example of smart surveillance systems for airport security is considered in [11]. This IBM S3, smart surveillance system, has two key components, namely, Smart Surveillance Engine (SSE) and Middleware for Large Scale Surveillance (MILS). SSE is responsible for performing event detection and supports video/image analysis. MILS supports the indexing and retrieval of spatio-temporal event meta-data. The example shown is the integration of many technologies like license plate recognition, behaviour analysis, face detection, and badge reading. However, the proposed system satisfies the requirements of *openness* and *extensibility*. However, the S3 framework has its own *airport system data model*, *user data model*, and *event data model*. Boll *et al.* [141] focused on the problem of analysing multimedia events for health monitoring. The proposed logical device layer-based architecture maps data from one or multiple (logical) devices into primary health features. Presently, the mapping for primary health features is canonical, i.e. scale directly delivers the values of body weight and fat. However, it could be extended for complex event processing like identification of "20 minutes cycling to work" using the time of day, GPS track, step counts, and past observations. An IoT based agricultural production system also proposed to analyse crop environments and improve the efficiency of decision making [227]. The designed system forecast agricultural production by monitoring crop growth periodically using IoT sensors. The system architecture can be divided into parts: relation analysis, statistical prediction, and IoT service. In statistical prediction, the production amount gets computed by estimating cultivated area and yield functions. It utilises text mining technology for relation analysis while analysing correlations of the agriculture-related text and locational conditions, selection, and replacement of crops. IoT service serves as an invaluable component that continuously monitors equipment and reports in real-time about the environment's conditions. Lastly, the design is implemented along with a GUI for visualisation.

Detection of interesting events automatically from broadcast sports video using object detection is one of the popular areas of event recognition [12, 228], utilizing the hierarchical structure of domain knowledge-based keywords. Similarly, surveillance-based

systems [11] designed for security events, satisfy principles of openness and extensibility, but only limited for various applications of security like homeland security, retail, manufacturing, mobile platform security, etc. Other domain-specific applications like flood detection, cultural event recognition, natural disasters, etc., are also introduced as a task in social media-based event recognition systems [13, 229, 230] with medium to high precision and no possibility for domain adaptation. Moreover, the detection and analysis of these natural calamities are also among major challenges of event recognition in remotely sensed data, and existing specialized methods are also more focused on delivering emergency response with high precision [14, 231, 232]. We can conclude that these multimedia event processing systems designed to achieve high performance, but most of them are domain-specific and cannot adapt to multiple domains.

Some of the recent works on object detection based multimedia events used for the comparison includes automatic vehicle detection and recognition for intelligent traffic surveillance system [233], firearm detection for security screening [234], unattended/stolen object detection by classifying objects as human or non-human [235], Car Parking Vacancy Detection [236] etc. Although these domain specific event recognition systems achieve high performance, they do not support a large vocabulary which limits their user interface, they also demonstrate the need to merge event based systems with multimedia methods with every change of domain, and therefore do not easily support domain adaptation.

It can be concluded existing multimedia based real-time systems possess high efficiency, but most of them are domain-specific. Moreover, lack of provision for the publish-subscribe paradigm [174] also limits the application of these systems in multiple scenarios of smart cities. Thus, merging event-based systems [8] with image processing systems is an essential requirement of current multimedia stream processing, which also limits the performance and requires high setup cost.

5.3.3 Multimedia Query Languages

Other than structured event query languages, we have plenty of video query languages like CVQL, SVQL, SPARQL-MM, VEQL [237–240], etc., proposed for processing image-based events. Content-based video query language (CVQL) is an extended version of existing structured query languages, that allows querying in video databases [238]. It is based on the spatial and temporal relationships of the content objects. Videos are divided into different classes, and then into their respective hierarchical categories. For example videos can be divided into *sports*, *politics*, *economics*, etc., and *sports* can be further classified into *basketball* and *tennis*. CVQL requires the knowledge of classes of

fixed domain, thus cannot fulfill the requirement of processing any type of unstructured events. Moreover, performance requirements are not addressed in this work. SVQL is based on the structure of sequential query language SQL with videos [239] by modifying the "where" clause to process them. It includes variable declaration, structure specification, feature specification and spatial-temporal specification in addition to the existing conditional expressions. With respect to requirements, SVQL is generalizable for handling unstructured events, however, execution methods have not been evaluated for efficiency and accuracy.

Similarly, SPARQL-MM [176] is also an extended version of SPARQL, by introducing spatio-temporal filter and aggregation functions for multimedia data. Although processing unstructured data is the primary goal of SPARQL-MM, it is not focused on its performance requirements. The goal of the MPEG Query Format (MPQF) is to facilitate and unify access to distributed multimedia repositories. MPQF can be used as a standard interface for multimedia retrieval engines [240]. It has the ability to access distributed multimedia repositories and uses an interactive feedback approach to improve efficiency which is not a recommended solution for real-time applications. The most recent Video Event Query language (VEQL) can express high-level user queries for video streams [237]. VEQL follows the SQL-like declarative expression and aims to use the standardized vocabulary of the existing event query language. It focuses on the spatial and temporal relationship among objects and their attributes.

While having the ability to deliver high performance, there is no discussion of including new classes or the possibility of adaptation in VEQL to include a large number of classes.

5.3.4 Gap Analysis

Table 5.1 summarizes the existing approaches with mapping of requirements (suggested in Section–2.4). While classifying the related work, we summarize the gap analysis with limitations as follows:

- *Event based Approaches for IoT*: Event-based approaches are mainly efficient in processing structured (scalar) events of smart cities. For instance, energy consumption, finance, packet loss events, etc., are more structured events as compared to multimedia events like traffic management, supervision, smart security, etc. Current event processing engines have no support for such unstructured event types. Moreover, there is no provision of incorporating multimedia query languages in event processing approaches of IoT. This shows the clear gap of lack of investigating and optimizing multimedia event processing in the current event-based middleware IoT approaches.

TABLE 5.1: Analysis of Related-Work with identified Requirements

Category	Approach	**Requirements**			
		High Accuracy for Multimedia Events	Low System Response Time	Support for Large Vocabulary	Maintain-ability
Event-based Approaches for IoT	SIENA [177, 179]	N.A	Best Effort	Application Specific	Expendable
	CEA [178]	N.A	Timely Response	Application Specific	Flexible
	SECO [180, 181]	N.A	Best Effort	Application Specific	N.E
	JEDI [182]	N.A	N.E	Application Specific	Flexible
	Hermes [186]	N.A	Best Effort	Reduced Semantic Gap	N.E
	STEAM [187–190]	N.A	Timely Delivery	Application Specific	N.E
	ToPSS [183, 184]	N.A	N.E	More Expressive Subscription Language	Support Changes
	S-ToPSS [185]	N.A	N.E	Semantic Matching Possible	General Approach
	OPS [191]	N.A	High Efficiency	Allow Se-mantically Equivalent Terms	N.E
	A-TOPSS [192–194]	N.A	High Efficiency	Support Approxi-mate Matching	N.E
	GREEN [195]	N.A	Hard Real-Time	N.E	Extensible
	EMMA [196]	N.A	Hard Real-Time	N.E	Autonomous
	Mires [197]	N.A	Non Real-Time	N.E	Not Supported
	SensorBus [198]	N.A	Non Real-Time	N.E	Not Supported
	MiSense [200]	N.A	High Through-put	N.E	N.E
	Rebeca [201–204]	N.A	High Efficiency	Limited Support	Manual
	PSWare [205]	N.A	Real-Time	High Ex-pressiveness	N.E
	TinyDDS [206]	N.A	Efficient	No Hetero-geneity	No Adaptation Possible
	Prisma [207]	N.A	Non Real-Time	N.E	Not Supported

N.E: Not Evaluated, N.A: Not Applicable

	Approximate Event Matcher [135, 209]	N.A	Efficient	Approximate Semantic Model	Manual
Application Specific Approaches for IoMT	Traffic Events [10, 222]	High Accuracy	Low Response Time for known Situations	N.A	N.A
	Sports Events [12, 228]	Limited Accuracy	Low Response Time for known Objects	N.A	N.A
	Security-based Events[11]	N.E	Low to Medium for known Objects	N.A	Extensibility
	Social Media based Events [13, 229, 230]	Medium Accuracy	N.E	N.A	N.A
	Satellite Imagery [14, 231, 232]	High Accuracy	Emergency Response for known Events	N.A	N.A
Multimedia Query Languages	CVQL [238]	N.A	N.E	N.A	Adapt with Domain Specific Knowledge
	SVQL [239]	N.A	N.E	N.A	N.E
	MPQF [240]	N.A	Efficient	N.A	Adapt on Feedback
	VEQL [237]	High Accuracy	High Through-put	N.A	N.E

N.E: Not Evaluated, N.A: Not Applicable

- *Application-Specific Approaches for IoMT*: Since existing real-time image processing systems are applicable for specific domains, we categorized them in application-specific approaches that possess the ability to process multimedia events for IoMT with high performance. As the name suggests, such approaches are not generalizable; developers have to build such multimedia applications every time with the change of domain by merging event processing and image processing systems. Ultimately, these approaches require high setup cost, high variance in performance, and not a unified user interface for the processing of IoMT based events. In conclusion, these approaches have high accuracy and low response time but no support for large vocabulary with limited maintenance facilities.

- *Multimedia Query Languages*: Most of these languages possess characteristics of detecting multimedia objects, supporting detection attributes, predicting spatial/temporal relationships, and efficient stream processing. Moreover, some of them are accurate and efficient but lack the ability of adaptation among domains to support large vocabulary.

5.4 Proposed Approach

Based on the approach of Information Flow Processing (IFP) systems [8], we proposed a Multimedia Event Processing Engine (MEPE) that consists of a matcher, Multimedia Event Processing Language (MEPL), Subscription Covering based Optimization, feature extraction and a collection of classifiers. In order to achieve the goal of performance requirements, a subscription-based optimization technique has been proposed along with MEPE. The feature extraction model within MEPE uses a DNN based approach to identify objects efficiently. Irrespective of providing high-performance capabilities in image recognition, DNN based techniques are also dependent on the trained classifiers for object detection. Conventionally, these classifiers are trained on general-purpose datasets consisting of a large number of classes, which may reduce the performance. The division of classifiers based on domain and selection of classifiers based on subscriptions could be a possible solution resulting in improvements in classifier performance. Thus, the proposed optimization is based on the inclusion of the "**classifier division and selection approach**", enabling the proposed feature extraction model to choose a suitable classifier based on subscription constraints for Hypothesis-I. The approach is presented with the implications of using only n-class classifiers belonging to a particular domain related to subscriptions with optimal values of "n" meanwhile neglecting classes of irrelevant domains.

5.5 Designing and Implementation

5.5.1 Generalized Multimedia Event Processing Engine

The proposed architecture for the multimedia event processing engine, along with its optimization for reducing the testing time, uses the following modules:

5.5.1.1 Receiver

Event Sources ("sensors" in the present case) create events, which get received by the *Receiver*, that sends events to MEPE for processing. The receiver implements the transport protocol to communicate information over the network [8]. It is also responsible for receiving information in the form of events from multiple sources and acting as an intermediator to send them one by one to the information flow processing (IFP) system, a multimedia event processing engine in the proposed method.

5.5.1.2 Multimedia Event Processing Engine (MEPE)

Matcher: The *matcher* is responsible for detecting conditions that hold in image events according to the user query (which has been evaluated using MEPL statements) and the propagation of notifications to the forwarder according to the condition detected in multimedia events.

Multimedia Event Processing Language (MEPL): Subscriptions are received by *MEPL Statement*, with "Image Event" from matcher, which analyzes the structure of the query and instantiates a feature extraction model while using *Subscription Covering based Optimization* for filtering commonalities. MEPL will be responsible for resolving the signatures of operators associated with the multimedia event based query languages such as the "Detect" operator in the present scenario described in Section–4.2.2.

Subscription Covering based Optimization: Subscription Covering based Optimization receives subscribed keywords with identifiers of subscribers from *MEPL*. It removes common keywords to consider them only once to further process multimedia events and sends the aggregated subscriptions to the *Feature Extraction* model. For instance, if multiple subscribers are looking for the same object (say "person"), then the keyword "person" should be analyzed once associated with numerous subscribers.

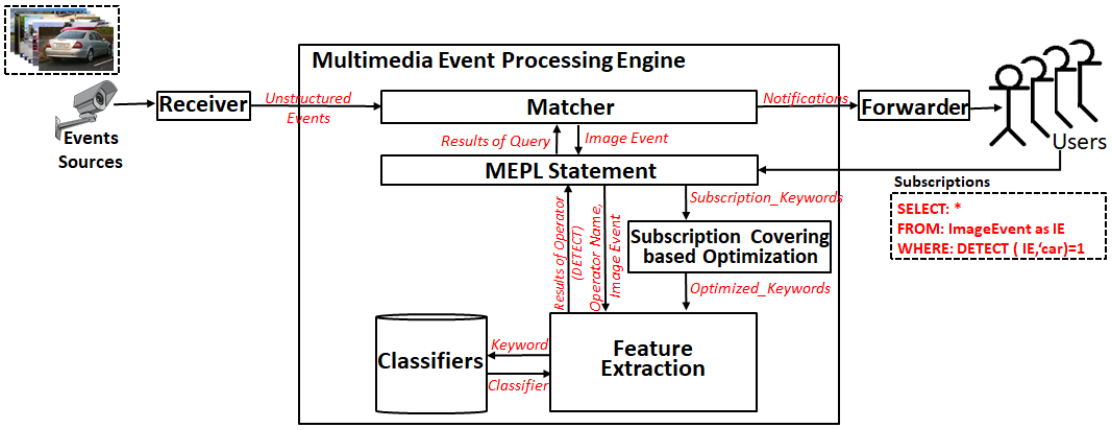


FIGURE 5.3: IoMT based Multimedia Event Processing model

Feature extraction: The feature extraction model performs operations on image events according to the subscriptions using image processing operators ("detect" operator in the present case) and a collection of *classifiers*. The DNN based feature extraction model is presently using "You Only Look Once" [35, 149] for object detection. Object detection is used to extract image features as it is the most common problem in the context of smart cities. Moreover, subscriptions from subscribers in the form of "keywords" will also direct the feature extraction model to use suitable classifiers to enhance performance according to classifier division and selection approach. This enables the use of subscription constraints for choosing the closely related classifier based on Hypothesis-I to process multimedia events.

The feature extraction model could also facilitates the proposed system to include multiple types of operators for processing different features of the multimedia events, which also makes it easily transportable to various domains and hence generalizable.

Classifiers: DNN based feature extraction model interacts with classifiers using keywords, which is a crucial requirement of the proposed classifier division and selection-based optimization methodology. Classifiers are trained on classes belonging to real-world objects to perform detection. The number of classes per classifier configuration may vary with a change of domains and will be responsible for the robustness of the system. Such N-Class classifiers of different domains provide us the settings for the proof of Hypothesis-I. The input "keywords" will direct a feature extraction model to choose the suitable classifiers for the processing of image events.

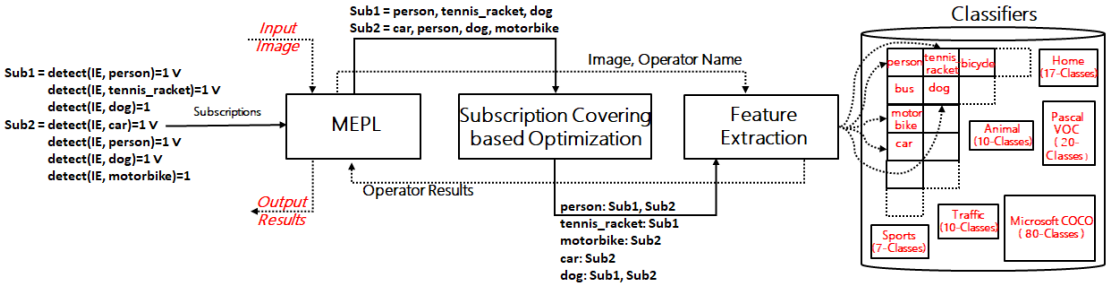


FIGURE 5.4: System Optimization using Subscriptions (Seen/Unseen Concepts)

5.5.1.3 Forwarder

Finally, events propagate to the forwarder, which will notify users according to their registered subscriptions. The "forwarder" is also necessary for the implementation of transport protocols for the purpose of communication.

Fig. 5.4 demonstrates the flow of subscriptions for the optimization of MEPE. Suppose registered subscriptions of users are $Sub_1 : \{detect(IE, person) \vee detect(IE, tennis\_racket) \vee detect(IE, dog)\}$ and $Sub_2 : \{detect(IE, car) \vee detect(IE, person) \vee detect(IE, dog)\} \vee detect(IE, motorbike)\}$. Subscriptions will be analyzed using MEPL, disintegrated into keywords, and communicated to the "feature extraction" model via "Subscription Covering based Optimization". The objective of prior subscription covering based optimization is to remove common *keywords* resulting in the detection of the only *person*, *tennis_racket*, *motorbike*, *car*, and *dog* as keywords for the feature extraction in the example. The feature extraction model will extract objects using an object detection model (presently "YOLO" [35]) using only specific classifiers related to the prescribed attributes (keywords). For instance "car" classifier (single class classifier) will be selected by the "YOLO" model for the detection of a *car*. The basic idea is to use only the available classifier, which is closely related to the attribute, which can vary from single to n-class classifiers having n ranges 1 to ∞.

Practically, it is not possible to design a classifier consisting of all probable classes (∞). Also, it is very unlikely that such a classifier would perform better than domain-specific classifiers. The utilization of n-class classifiers related to the particular domain may lead to improvements in performance compared to the use of a general-purpose classifier with a large number of classes. For further illustration, we are also considering the traffic, sports, animal, and home subscriptions related to classifiers having 8, 9, 10, and 17 classes, respectively derived from Pascal VOC [26] and Microsoft COCO [27] datasets. Since the constructed n-class classifiers belong to different domains, they also comply

with the goal of generalizability. Lastly, MEPE classifiers also include two general-purpose classifiers with 20 and 80 classes constructed using Pascal VOC and Microsoft COCO for the proof of optimization described in Section–5.6.3.2.

5.5.2 Multimedia Event Processing Algorithms

The utilization of DNN based systems for high performance requires a classifier selection technique for analyzing multimedia events. Thus, we present a method that can analyze multimedia events only once to process multiple subscriptions. Besides, an optimization technique, "subscription-based classifier selection," for finding classifiers is proposed in this section. The implementation procedures for optimized IoMT based model and handling of commonalities among subscriptions are shown in Algorithm 1 and 2, with descriptions of symbols in Table 5.2.

5.5.2.1 Multimedia Event Processing Engine

The proposed multimedia event processing algorithm (Algorithm 1) uses the following four steps and is based on keyword-based optimization for handling commonalities:

1. Finding commonalities among subscriptions for subscription covering based optimization using keywords.
2. Identification of classifier according to the subscribed keyword.
3. Application of object detection for the processing of image events.
4. Notification of user on the matching of image event with a subscription.

Let *List(L)* consist of sets of subscriptions (Sub_i) with each set consisting of the identity of subscribers (S_i) with a set of keywords ranging from 0 to K_{j_i}. Consider a stream of image events as S_{IE}. The algorithm starts with the distribution of subscriptions into a two-dimensional adjacency matrix $T(K, S)$ with K as keywords and S as indexes of subscribers corresponding to keywords, using the procedure of handling commonalities. Initialize m and n with the total number of keywords and subscribers, respectively. The algorithm keeps on receiving the image events as IE and process them for each keyword k_i belonging to $T(K, S)$ with i ranging from 1 to m. Each iteration begins with the identification of *class name* related to keyword k_i and recognition of specific *classifier* using the class name. Thereafter, objects are extracted from image events using the *matcher*, and users are notified on the matching of class names with objects detected.

TABLE 5.2: Description of Symbols

Symbol	Description
$List(L)$	List (L) of sets of Subscriptions
$T(K,S)$	Two-dimensional matrix for indexing Keywords and Subscribers
K	Keywords
S	Subscribers
S_{IE}	Stream of Image Events

Algorithm 1 : Multimedia Event Processing Engine

Input: List (L) of sets of Subscriptions:
$(Sub_1 : \{S_1, K_{1_1}, K_{2_1}, ..., K_{j_1}\},$
$Sub_2 : \{S_2, K_{1_2}, K_{2_2}, ..., K_{j_2}\}, ...$
$Sub_i : \{S_i, K_{1_i}, K_{2_i}, ..., K_{j_i}\})$ and
S_{IE}: Stream of Image Events.
Output: Notifications

T(K,S)$\leftarrow SCBO(L)$ {Subscription_Covering_based_Optimization}
$m \leftarrow count\_keywords(T(K,S))$
$n \leftarrow count\_subscribers(T(K,S))$
while true **do**
  $IE \leftarrow Image\_Event(S_{IE})$
  for $i = 1$ to m **do**
    $classname \leftarrow \{k_i |\ k_i \epsilon T(K,S)\}$
    $classfier \leftarrow find\_classifier(classname)$
    $objects \leftarrow matcher(IE, classifier)$
    if $(classname \in objects)$ **then**
      for $j = 1$ to n **do**
        if $(T(k_i, s_j) = 1)$ **then**
          $notify(s_j)$
        end if
      end for
    end if
  end for
end while

5.5.2.2 Subscription Covering based Optimization

Algorithm 2 shows the steps of keywords based optimization used for handling commonalities among subscriptions. Keywords based optimization is the procedure of identification of common keywords among all subscriptions and considering them only once for the processing of image events. The procedure of handling commonalities consists of the following major steps:

1. Instantiate keyword-subscriber two-dimensional matrix using input subscriptions.
2. Identification of keywords for each subscriber.

3. Association of subscriber identities with an index of requisite keywords.

Let the *List(L)* be the input, and $T(K, S)$ be the output two-dimensional adjacency matrix, consisting of keywords and subscribers identifiers. Next, initializes the total number of subscribers as i, counting the total number of keywords as j and assign $T(K_{j_i},\ S_i)$ as 1 with each keyword K_{j_i} belonging to a subscriber (S_i) using subscription set (Sub_i).

Algorithm 2 : Subscription Covering based Optimization

Input: List (L) of sets of Subscriptions S:
$(Sub_1 : \{S_1, K_{1_1}, K_{2_1}, ..., K_{j_1}\},$
$Sub_2 : \{S_2, K_{1_2}, K_{2_2}, ..., K_{j_2}\}, ...$
$Sub_i : \{S_i, K_{1_i}, K_{2_i}, ..., K_{j_i}\})$
Output: Decision Tree

```
T(K,S) ← 0
i ← Count_Subscribers(S)
for 1 to i do
    j ← Count_Keywords(Sub_i)
    for 1 to j do
        T(K_{j_i}, S_i) ← 1
    end for
end for
return T(K,S)
```

5.6 Evaluation

5.6.1 Evaluation Methodology

Based on traditional evaluation metrics, I am using throughput and accuracy to evaluate the efficiency and effectiveness of the multimedia event-based system. In addition to these performance metrics, I also compare the single-class classifiers with existing n-class classifiers using precision and recall for the proof of optimization. To test Hypothesis-I, I perform experiments on different values of "N" of N-class classifiers while measuring their throughput (for response-time) and accuracy. Moreover, I use the N-class classifiers of multiple domains to prove the ability to generalize. Specifically, experiments have been conducted on randomly generated subscriptions by considering multiple users ($m\epsilon[1, 10]$) and a large number of subscriptions ($n\epsilon[1, 100]$) per user with following experimental setup:

- **Event Sets**: Five event sets consisting of events related to traffic, sports, home, animal, and mixed events have been prepared manually using classes of the Microsoft Common Objects in Context (COCO) dataset [27]. Among 80 classes of

Microsoft COCO, I found 8 classes related to traffic, 17 classes related to home, 9 classes related to sports, and 10 classes related to animal. For instance, I construct traffic classifier (i.e., 8-Class classifier) from training data of classes like person, bicycle, motorbike, bus, truck, traffic light etc. Then I used the testing data of the same classes for the generation of testing events which are $\approx$ 25818 for traffic events. Similarly, I construct an animal classifier (i.e., 10-Class classifier) using training data of classes like bird, cat, dog, horse, etc. I prepare animal events using images available in testing data with at least one or more classes related to bird, cat, dog, horse, etc. I use the same approach to construct classifiers and testing events for sports and home category. Then I prepare a mixed event based classifier where I consider all classes of traffic, sports, home, and animal classifiers. It is important to note that I prepared only categories traffic, sports, home, and animal because Microsoft COCO consists of only 80 classes, and this was the most suitable classification.

- **Subscription Sets**: Random subscriptions have been generated manually for each of the applications by varying the number of users and number of subscriptions per user. Subscriptions consist of attributes having the name of objects for detection like $\{cat\}$, $\{dog\}$, $\{cat, dog, horse\}$, $\{car, motorbike\}$, etc.

- **Matcher Constraints**: Given a set of n subscriptions $S = \{s_1, s_2, ..., s_n\}$ with collective attributes $A = \{a_1, a_2, ..., a_t\}$ and a set of classifiers $C = \{c_1, c_2, ..., c_m\}$, the event matcher has to match events with subscriptions S using classifiers C. The matcher constraint is a condition specified on set of attributes A that belongs to subscriptions S. There exists a subset C' of available classifiers C, which covers all attributes of A, *i.e.*, $\exists\ C'\ \forall(a_i\ \epsilon\ A)(C' \subseteq C \wedge a_i\ \epsilon\ C')$, to achieve high performance.

To derive the trend of throughput with number of classes, and for the proof of optimizations, I use the below training and testing datasets:

- Training Datasets: All 80 classifiers (classifier with 1 class, the classifier with 2 classes, the classifier with 3 classes, and so on) are trained on Microsoft Common Objects in Context (MCOCO) training dataset to demonstrate the dependence of throughput on the number of classes. Also, two other specialized classifiers are trained on Pascal VOC [26] and Microsoft COCO [27], having 20 and 80 classes, respectively, to compare precision and recall.

- Testing Datasets: I use the testing dataset of Microsoft COCO [27] to analyze throughput with the number of classes because Microsoft COCO (80 classes) has more classes than Pascal VOC (20 classes). However, I use both Pascal VOC

and Microsoft COCO testing datasets to show the precision and recall for N-Class classifiers.

All experiments have been conducted on Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0-26-generic x86_64), with NVIDIA TITAN Xp GPU.

5.6.2 Evaluation Metrics

- Throughput: Number of events matched in a unit time, measured in terms of frames/sec (fps).
- Accuracy: Ratio of correctly predicted observation to the total observations.

$$Accuracy = (TP + TN)/(TP + FP + FN + TN) \tag{5.1}$$

- Precision: Ratio of correctly predicted positive observations to the total predicted positive observations.

$$Precision = TP/(TP + FP) \tag{5.2}$$

- Recall: Ratio of correctly predicted positive observations to the all positive observations of actual class.

$$Recall = TP/(TP + FN) \tag{5.3}$$

5.6.3 Experiments and Results

5.6.3.1 Evaluation of Feature Extraction

Table-5.3 summarizes the average accuracy and throughput of the proposed system and demonstrates overall performance improvement for analyzing traffic, sports, home, animal, and mixed image event streams. The accuracy of the proposed method is measured with the help of correctly predicted observations (true positives and negatives) with respect to the total number of observation/image events. In contrast, throughput indicates the number of frames processed per second (fps). Performance has been evaluated for each type of event stream using three types of classifiers: single class, N-class, and 80-class classifiers, to demonstrate the impact of an increased number of classes per classifier on performance. Here the value of "N" depends on the application domain, which is taken as 8, 9, 17, and 10 for traffic, sports, home, and animal classifiers, respectively. The 80-class classifier serves the purpose of a general classifier having Microsoft COCO classes consisting of multiple domain categories [27], and it will remain the same throughout all experiments.

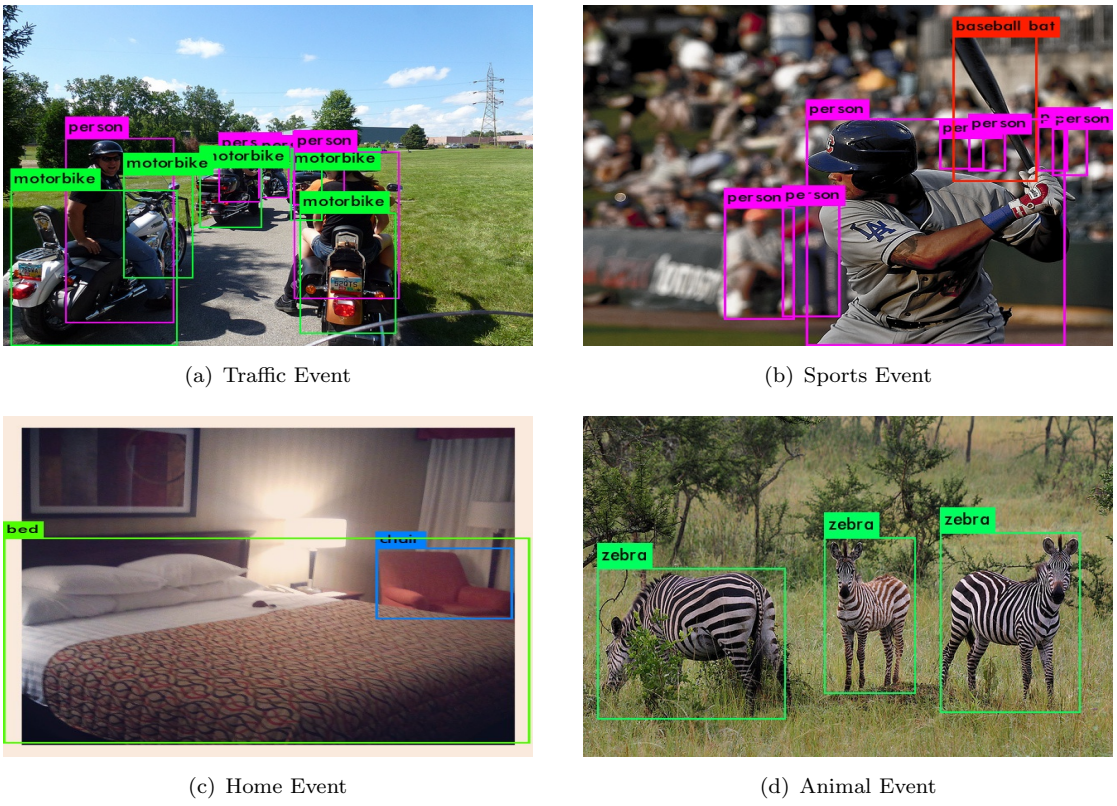
Additionally, performance has been evaluated by replacing a single 80-class classifier with category related single-class classifiers. Results show that the throughput of the system is increased while having competitive accuracy. The only drawback of using single-class classifiers is the requirement of loading all classifiers at the same time. However, the decrement in throughput and accuracy on using 80-class classifier compared to single class classifiers shows that the system's performance will decrease with an increase in the number of classes per classifier. Thus, it will be beneficial to choose the optimal value of "n" and consider only the related classes for constructing a classifier for optimization.

Consequently, we also evaluated our system on domain-specific (N-Class) classifiers that consistently outperform the other classifiers with an average throughput and accuracy of 110 fps and 66.34%, respectively. This verifies our Hypothesis-I. The high average throughput of the proposed system consisting of N-Class classifiers for different event streams while achieving high accuracy signifies the generalizability, efficiency, and effectiveness of the proposed design for real-time applications.

The performance of the *detect* operator on multiple types of events for handling multiple domain-related subscriptions is shown in Fig. 5.5.

TABLE 5.3: Performance of proposed MSPE on different classifiers in terms of Accuracy and Throughput (FPS)

Exp. No.	Events	Example of Subscriptions	Accuracy				Throughput (fps)			
			Proposed Approach			Overall Change	Proposed Approach			Overall Change
			80-Class Classifiers	1-Class Classifier	N-Class Classifier		80-Class Classifiers	1-Class Classifier	N-Class Classifier	
1.	Traffic Events ≈ 25818	person, bicycle, car, motorbike, bus, truck, traffic light, stop sign, etc.	16.34%	94.58%	16.35% (N = 8)	+ve (0.06%)	108.08	114.67	111.18 (N = 8)	+ve (2.87%)
2.	Home Events ≈ 17218	chair, sofa, potted plant, bed, dining table, toilet, tv, refrigerator, etc.	83.15%	74.93%	83.19% (N = 17)	+ve (0.05%)	107.83	114.04	110.20 (N = 17)	+ve (2.20%)
3.	Sports Events ≈ 7776	frisbee, snowboard, sports ball, baseball, bat, skateboard, tennis racket, etc.	90.26%	68.04%	90.26% (N = 9)	No Change (0.00%)	107.72	114.94	110.65 (N = 9)	+ve (2.72%)
4.	Animal Events ≈ 8265	bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe, etc.	86.51%	68.60%	86.45% (N = 10)	-ve (0.07)%	108.01	114.56	110.81 (N = 10)	+ve (2.59%)
5.	Mixed Events ≈ 40137	person, aeroplane, boat, bottle, wine, glass, book, glove, surfboard, cat, etc.	49.45%	81.72%	55.45% (N = 44)	+ve (12.13%)	106.23	114.55	107.16 (N = 44)	+ve (0.88%)



(a) Traffic Event

(b) Sports Event

(c) Home Event

(d) Animal Event

FIGURE 5.5: Detect Operator on Traffic, Sports, Home and Animal related events

5.6.3.2 Proof of Optimization

Throughput vs Number of Classes Fig. 5.6 illustrates the relationship of throughput with the number of classes for n-class classifiers, where n ranging from 1 to 80. Here, n-class classifiers refer to a classifier that is trained on n classes. The performance of the system on using single-class classifiers (classifier trained on only one class) will remain constant, with the condition of loading of all classifiers at the same time.

Also, it can be seen from the graph that the throughput of n-class classifiers is continuously decreasing with an increase in the number of classes, which validates our Hypothesis-I in terms of low response-time. This shows that the number of classes is among one of the configuration parameters of classifiers, which affects their time complexity with loose upper bound $o(n)$. Thus choosing an optimal value of n according to the required application is recommended for the optimized solution of constant time complexity $\mathcal{O}(1)$.

Precision-Recall vs N-Class Classifiers Fig. 5.7 represents the comparison of average precision and recall for three types of classifiers: Pascal VOC, Microsoft COCO, and single-class trained classifiers. Here a Microsoft COCO trained classifier with 80

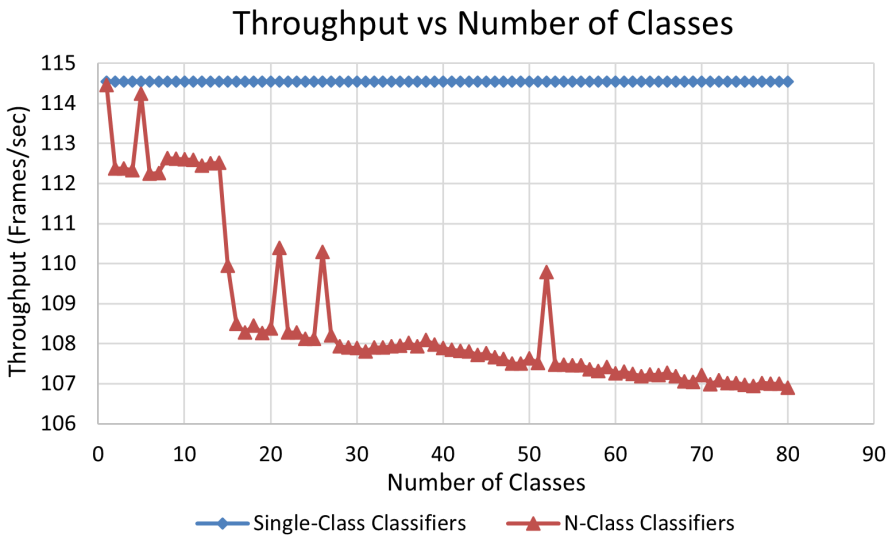
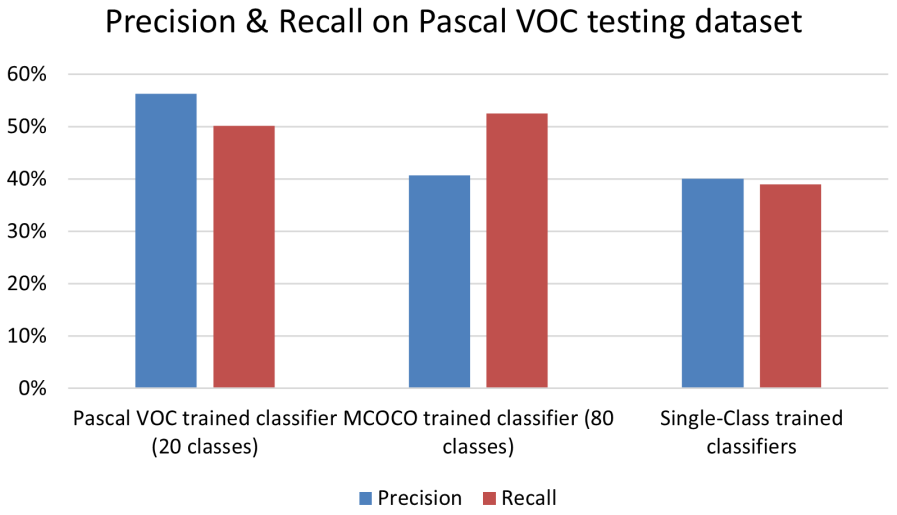


FIGURE 5.6: Average Throughput of Classifiers

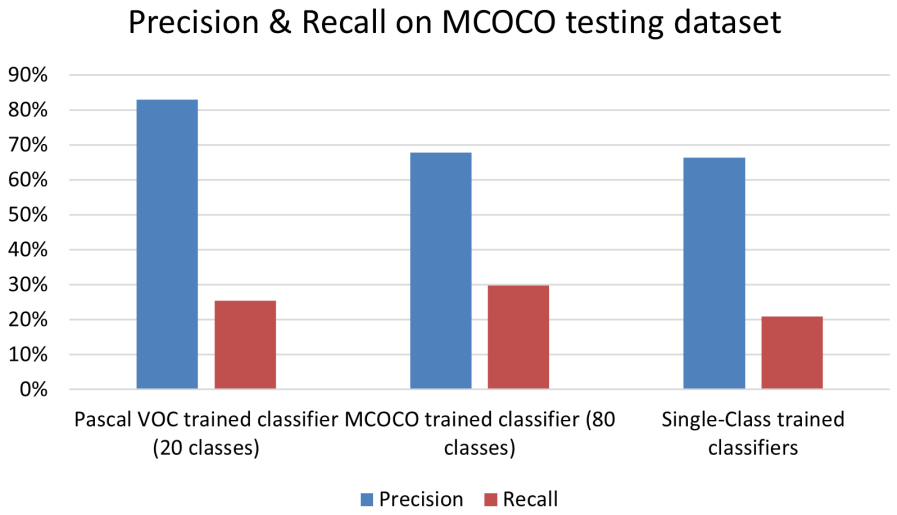
classes serves the purpose of testing the best case analysis as compared to a single class classifier trained on 1 class for the worst-case analysis. However, Pascal VOC with 20 classes is considered to analyze the performance on average cases. Fig. 5.7(a) compares the performance of all types of classifiers on the Pascal VOC testing dataset. It shows that the Pascal VOC trained classifier is performing best on the Pascal VOC testing dataset. Simultaneously, the precision-recall of Microsoft COCO with 80 classes is almost equivalent to the performance of Pascal VOC having 20 classes. However, the values of precision and recall achieved by single-class trained classifiers, which could be the worst possible case of N-class trained classifiers, are also quite promising compared to other classifiers. The values of precision and recall also support our Hypothesis-I for accurate multimedia event processing. Similarly, Fig. 5.7(b) shows the performance of all three types of classifiers on the Microsoft COCO testing dataset. The competitive precision-recall of the approach having N-class classifiers with N=1 is practically indistinguishable compared to N=20 and N=80 classes, which also indicates the suitability of keeping the value of N as minimal as possible.

It implies that choosing lower values of "N" related to the application domain can improve the system's throughput and does not influence its accuracy. Hence the "classifier division and selection" approach has proven to be useful for the purpose of optimization of feature extraction.

Other than these experiments, I believe performance may change with the size of images and quality. For instance, Pascal VoC and Microsoft COCO are benchmark object detection datasets, thus consisting of more objects per image with many boxes with good resolution for the training. However, in a real-time environment, we may get iconic



(a) Average Precision and Recall on Pascal VOC testing dataset



(b) Average Precision and Recall on MCOCO testing dataset

FIGURE 5.7: Average Precision-Recall of Classifiers

images of specific objects for testing/training that may increase or decrease the accuracy. However, the trends of decrease in throughput with the increase in the number of classes demonstrated in experiments will remain the same. Availability of a large number of images for training may also improve the performance of Model-I.

5.7 Conclusion and Discussion

To process multimedia events within the event-based paradigm, a generalizable IoMT based system has been proposed in this chapter. Literature reveals that existing event processing approaches in IoT do not support multimedia events, while image processing approaches are application-specific. The proposed approach is based on a deep neural

network-based feature extraction model expressed as an object detection operator. The proposed model has been optimized by using a classifier selection approach based on subscription constraints. Experiments show that the proposed system achieves an average throughput of 110 frames/sec with an approximate accuracy of 66.34% on real-world events in various applications of smart cities. We show the decrease in classifiers' performance with an increase in the number of classes per classifier, which indicates the effect of a number of classes on the performance of the proposed system by time complexity $o(n)$. Precision and recall have been evaluated to show the reasonable performance of n-class classifiers even in a worst-case scenario. The reduction in throughput with an increase in the number of classes and promising precision and recall even on small values of n for n-class classifiers supports choosing the optimal number of classes per classifier to achieve constant high performance. Thus, it is evident that the proposed approach is capable of providing the desired optimization based on classifier configuration using subscription constraints, which verifies Hypothesis-I.

Presently our proposed approach requires trained classifiers for the processing of unseen concepts. Thus, it can extend the proposed model with online training with the introduction for the self-construction of classifiers in the future. Such a direction could lead to the inclusion of adaptability (Chapter–6) in the sense that the system can be generalized or specialized according to the practicality of the model in various scenarios.

Chapter 6

Hyper-Parameters based Adaptive Multimedia Event Detection

6.1 Introduction

The goal of generalized multimedia event processing [32] was analyzed in previous Chapter-5, and I reached towards a key open challenge of trained classifiers availability for the processing of multimedia events using neural network-based techniques in real-time. Current online learning approaches make their decisions on the fly [19, 20]. Still, they are only based on concept drift in multimedia streams, and inapplicable for the handling of new/unknown subscriptions belonging to multiple applications of smart cities. Apart from the limitation of availability of pre-trained classifiers, the optimisation techniques in neural network models are based on the trade-off of speed and accuracy [22], which is supposed to be done before the processing of events and cannot be configured at run-time in case of adaptive subscriptions of multiple domains. Therefore, there is the requirement for an online classifier construction-based approach, that can answer seen/unseen subscriptions by processing multimedia events with minimal response time and high accuracy.

The choice of hyperparameter values [241] greatly affects the performance of resulting classifiers and could be a possible solution for reduced response-time. In this chapter we tests the research Hypothesis-II "*If tuning of hyperparameters based technique is useful in machine learning models to speed-up the training, decrease the computation cost, and increase the accuracy; then performance will get enhanced for low response-time also even on training from scratch for unseen subscriptions on tuning hyperparameters for*

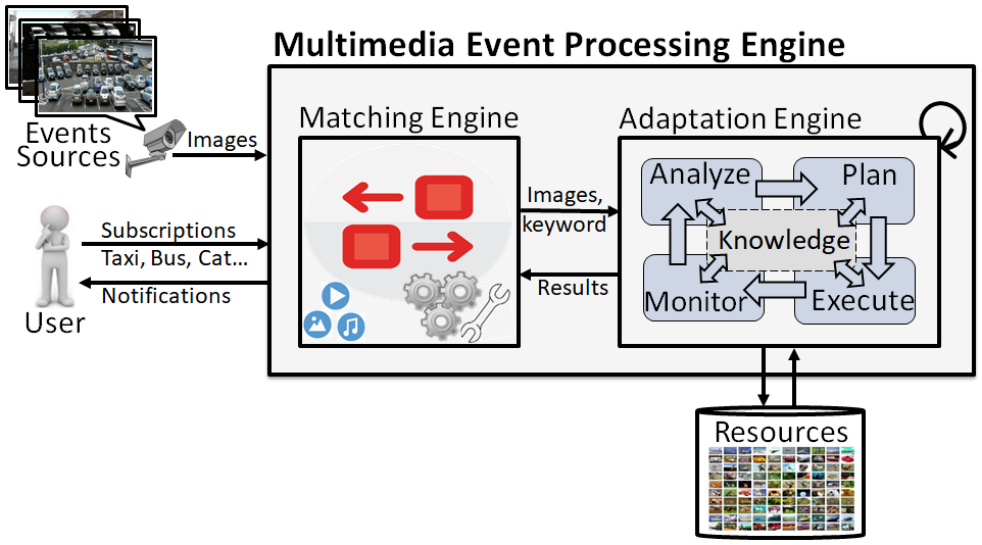


FIGURE 6.1: Conceptual Architecture for the Online Adaptive Classifier based Multimedia Event Processing

the online construction of classifiers.". Before testing the hypothesis, we formulate this problem into our Research Question 2 as "*How can we answer multimedia event based queries online consisting of completely unseen subscriptions (unbounded vocabulary), using an adaptive classifier construction approach with the tuning of hyperparameters while achieving high accuracy and minimizing the response time?*" in Section–6.2.

Specific existing research related to the online training and self-adaptation of classifiers is presented in Section–6.3. To achieve high-performance multimedia event processing, publish-subscribe based systems are incorporated with online classifier learning-based neural network models specifically to detect objects (detailed in Section–6.4 and 6.5). The multimedia stream processing engine allows users to subscribe to classes belonging to any domain, monitor multimedia events, and process them using an event-based matcher, adaptation model, and classifier based object detection models (shown in Fig. 6.1). We optimise the multimedia stream processing model with a self-adaptation model that analyses the accuracy-processing time trade-off of object detection models at run-time and configure it using performance-based strategies on dynamic subscriptions. We leverage hyperparameter tuning-based techniques, including the configuration of *learning-rate*, *batch-size*, and the *number of epochs* for the optimisation. We consider mainly three strategies: *Minimum Response Time needed while Minimum Accuracy allowed*, *Optimal Response Time needed while Optimal Accuracy allowed*, and *Maximum Response Time allowed while Maximum Accuracy needed*, for the requirement of high performance in multimedia event processing applications.

Our experiments (presented in Section–6.6) demonstrate that deep neural network-based

object detection models, with hyperparameter tuning, can improve the performance within less training time to answer previously unknown user subscriptions. This study shows that the proposed online classifier training based model can achieve accuracy of 79.00% with 15-min of training and 84.28% with 1-hour training from scratch on a single GPU for the processing of multimedia events. Lastly, Section–6.7 concludes and discusses the limitations with the need for inclusion of domain adaptation.

6.2 Problem Overview

6.2.1 Preliminaries

- Hyperparameters: Hyperparameters are configuration parameters of the model that cannot be trained directly from the training data, and often specified by practitioners after resort to experimentation's. Examples of hyperparameters may include learning rate, number of epochs, batch size, number of hidden layers, architecture, activation functions, etc.

- Online Learning: Precisely, *online learning* is answering a sequence of questions given knowledge of the correct answers to previous questions and possibly additional available information [242]. Online learning-based approaches are adaptable, make their decisions on the fly, and applicable for situations in which data changes frequently. Due to the term "online learning", the standard approach to machine learning got the name "offline learning", where we use a source dataset and train a model on the whole dataset at once. This offline learning is often called *batch learning* [243], as most models get trained in a batch manner. In an offline case, if the model needs to learn about new data, models need to be retrained on new data. On the other hand, training happens incrementally in *online learning*.

- Self Adaptation: It refers to automate the procedure of configuration of models or self-acting on them in case of reduced accuracy of real-time applications. From the machine learning perspective, these are automatic selection methods for algorithms or hyperparameter values for a given supervised problem [244]. The goal of these methods is to quickly find the effective algorithm and/or combination of hyperparameter values that maximizes the accuracy within a pre-specified resource limit, where resource limit mainly includes the amount of training time, number of values to be tested, or number of scans over data. Auto-Weka [245] and Hyperopt-Sklearn [246] are two popular examples of automatic configuration approaches.

- TPE: Like other Sequential model-based optimization (SMBO) algorithms, Tree-structured Parzen estimators (TPE) also differs in surrogate model $p(y|x)$, which

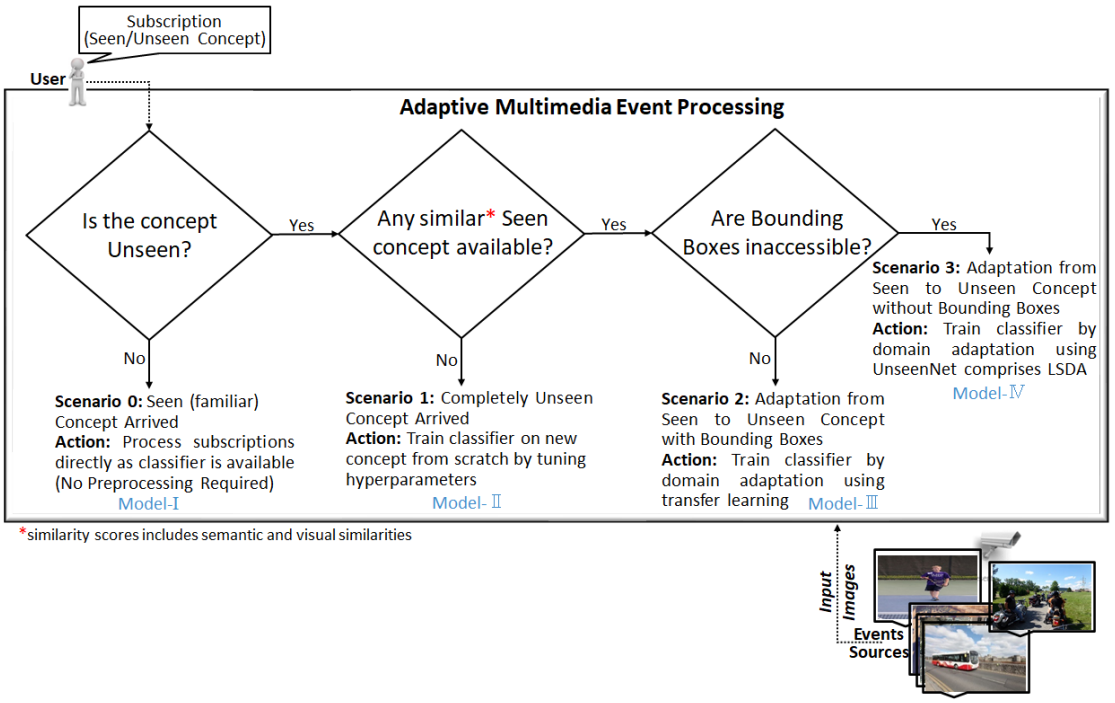


FIGURE 6.2: Scenarios for Multimedia Event Processing adhering to Seen/Unseen Concept Problem

is the probability of the hyperparameters. The TPE [241] defines $p(x|y)$ using two such densities:

$$p(x|y) = \begin{cases} l(x) & if\, y < y^* \\ g(x) & if\, y \geq y^* \end{cases}$$

where $l(x)$ is the density formed by using the observations such that corresponding loss was less than y^* and $g(x)$ is the density formed by using the remaining observations.

6.2.2 Motivational Scenarios

Consider the baseline scenarios of detecting seen/unseen concepts for analyzing multimedia events, as shown in Fig. 6.2. As we have seen in the previous chapter, we can process multimedia events directly from the pre-trained classifiers for any seen concept. However, if the concept "unseen" completely, i.e., there is no similar seen concept-based classifier available in the multimedia event processing model for the knowledge transfer. In that case, we need to address the problem of classifier training for unseen concepts from scratch in low response time. This specific problem associated with Scenario-1 is described in Fig. 6.3. If a user subscribes for *mirror* detection and existing public traffic control management system (having *bus*, *car*, *traffic-light*, *bicycle*), security management

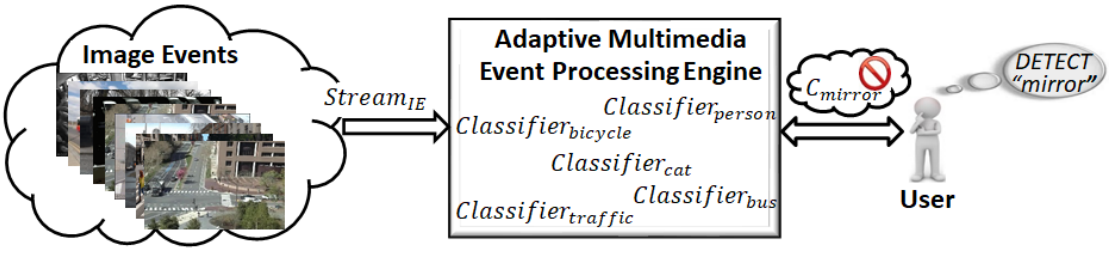


FIGURE 6.3: Scenario-1: Completely Unseen Concept Arrived (Application of Model-II: Hyper-Parameters based Adaptive Multimedia Event Detection)

systems (having *person*, *building*, *key*), smart home systems (having *sofa*, *television*, *utensils*, *table*) *etc* can recognize only their limited classes. Then such system may require manual effort for the answering of any completely unknown subscription "mirror". However, with the provision of online training for the handling of new subscriptions in reduced response-time, such types of queries can be answered automatically by training a new *mirror* classifier while using deep neural network-based models, thereby eliminating the sudden breakdowns that existing systems currently exhibit in such scenarios.

6.2.3 Problem Statement

The problem can be defined more specifically as "*How can we answer multimedia event based queries online consisting of completely unseen subscriptions (unbounded vocabulary), using an adaptive classifier construction approach with tuning of hyper-parameters* ($\vec{\lambda}$, $\vec{\mathcal{E}}$, $\vec{\beta}$) *while achieving high accuracy and minimizing the response time?*".

6.3 Background and Related Work

6.3.1 Online Learning of Classifiers

Online learning is the branch of machine learning useful in environments where data behaviors change quickly, like shipping websites, product search, stock price prediction, etc [242]. Online learning based approaches make their decisions on the fly. Thus such machine learning-based approaches could prove to be useful for the adaptation among classifiers. Online learning based algorithms are scalable and data-efficient that learn to update models from data streams sequentially, and no longer require data which has been consumed [117, 247]. Data streams frequently experience "concept drift" as a result of changes in the underlying concepts. When data distribution drifts due to changing behavior of customers, online learning models can adapt on-the-fly to keep pace with trends in real-time. This procedure is similar to offline learning, where we create a

TABLE 6.1: Analysis of Related-Work with identified Requirements

Category	Approach	**Requirements**			
		High Accuracy for Multimedia Events	Low System Response Time	Support for Large Vocabulary	Maintainability
Online Learning of Classifiers	Active Learning [248–250]	High Accuracy	N.E	Scalable Dataset Construction Support	High Maintainability
	Semi-Supervised Learning [251–253]	High Accuracy	N.E	N.A	Adaptive within Domain
	Adaptive Classifier Learning [19–21, 254, 255]	Average to High Accuracy	Low Training Time	N.A	High Maintainability
Self-Tuning of Classifiers	Hyperparameter Tuning [241, 244, 256–258]	High Accuracy	N.E	N.A	N.A
	Self-Adaptation [23, 24]	High Accuracy	N.E	N.A	Adaptive for Performance Metrics

N.E: Not Evaluated, N.A: Not Applicable

sliding window of data and retrain it every time. Online learning-based approaches are adaptable and can easily handle concept drifting of data streams, which makes this methodology crucial for streaming analytics [19].

Most of the existing techniques of online learning are based on semi-supervised or active learning of classifiers. Active learning is one of the techniques that allow the machine learning methods to select a subset of the unlabeled data from the data distribution to be labelled [248, 249]. Uncertainty sampling [259], Query-by-Committee (QBC) [260] and Estimation of error reduction [250] are the most popular methods to perform active learning. Although active learning is an enhancement over conventional inductive learning; the approach requires the construction of an exhaustive labelled dataset which is laborious and challenging [261]. Semi-Supervised learning is also another step towards online learning which requires a small amount of labelled data as compared to unlabeled data [251]. These algorithms [252, 253] work on the hypothesis that the labels generated by the base learner with high confidence can be added to the training dataset, and able to improve the accuracy. Although the accuracy of such existing methods is relatively

high with high speed of stream processing, given that these methods are semi-supervised, they cannot handle multimedia streams online.

Existing adaptive classifier based machine learning techniques [20, 21, 254, 255] in this category are designed with the aim of evolution of classifiers with drift in concept of multimedia streams. Wang et al. [20] proposed a framework for concept drifting of data streams using weighted classifier ensembles. However, the identification of *concept drift* in these dynamic approaches is mainly focused on the processing of structured data streams and cannot accommodate multimedia data streams. An ensemble of classifiers is based on combining the results of individual classifiers and producing more accurate results for dynamic data streams, thus suitable for online learning [262]. The approach is based on dynamic classifier ensembles but more focused on structured event streams for analyzing concept drift. However, it has been investigated that dynamic ensemble selection scheme performs better than static ensemble selection in some cases [263]. Other recent adaptive classifier based techniques [254, 255] are efficient but applicable only for particular domains. The identification of *concept drift* in these dynamic approaches, is mainly focused on processing of structured data streams and cannot accommodate multimedia data streams. Online learning can be directly applied to deep neural networks, but also they suffers from convergence issues and forgetting previously learned data [264].

Although online learning-based approaches remove the constraint of availability of classifiers, most of them are solely based on concept drift in multimedia streams and thus become inapplicable for handling dynamic subscriptions. Moreover, handling the challenge of changing/inconsistent interest of the user, and adapting classifiers accordingly, need to be investigated for the generalized framework of multiple domain-based streams [265].

6.3.2 Self-Tuning of Classifiers

Hyperparameters are configuration parameters of the model that cannot be trained directly from the training data, and often specified by practitioners after resort to experimentations. Examples of hyperparameters may include learning rate, number of epochs, batch size, number of hidden layers, architecture, activation functions, etc., and choosing the right set of these values is typically known as *Hyperparameter tuning*. Since the choice of hyperparameter values greatly affects the performance of resulting classifiers, various automatic selection methods were proposed in the literature [244] for

hyperparameter values. The most common algorithms for the selection of hyperparameters are ranging from Grid Search, Random Search, Bayesian Optimization, Sequential Model-Based Optimization to Tree-structured Parzen Estimators (TPE) [241].

Grid search is the simplest method of hyperparameter tuning. It is a brute force method that trains the model for all combinations of parameters specified in a grid and selects hyperparameters after evaluating each model. Since grid search suffers from having high dimensional space, it is computationally very expensive. Random search is different from a grid search as it assumes that not all hyperparameters are equally important. In this method, we provide the statistical distribution for each hyperparameter from which values may randomly sample. We may also define the total number of iterations, and the hyperparameter values of the model will be set and evaluated for each iteration from a specified probability distribution. As compared to the grid search, the random search has much improved exploratory power [256]. Bayesian approaches keep track of the previous iteration results to improve the sampling method for the next experiment [257]. There are two main decisions that we need to make for Bayesian optimization: (1) Selecting a prior over functions to express assumptions about the function being optimized (for instance Gaussian Process prior) (2) Choosing an acquisition function to determine the next point to evaluate. Sequential model-based optimization (SMBO) algorithms formalize Bayesian optimization [241]. It iterates between fitting models sequentially while trying each time better hyperparameters using Bayesian reasoning and updating the probabilistic model.

Many variants of SMBO algorithms exist which differ only in the surrogate model, where the surrogate is the model used for approximating the object function. TPE builds a surrogate model by applying Bayes's rule [258]. This method is restricted only to tree-structured configuration spaces, i.e. leaf variables only make sense when node variables take particular values. TPE first samples the hyperparameter search space by random search, then it divides the output scores into two groups. The first group consists of best scores and the second group contains the rest of the observations by assuming y^* as the splitting value for the two groups. Then the two densities $l(x)$ and $g(x)$ are modelled using Parzen Estimators, where $l(x)$ and $g(x)$ are averages computed from kernels centred on existing data points. The TPE algorithm defines likelihood probability as $p(x|y) = l(x)$ if $y < y^*$ or $p(x|y) = g(x)$ if $y \geq y^*$. The model evaluates the sample hyperparameters according to $l(x)/g(x)$, updates observation list, and iterates over a fixed number of trials. The major advantage of the TPE is that it allows a vast domain for hyperparameter search space. These baseline methods are further integrated into open source softwares for the automatic selection of algorithms and hyperparameter values.

These baseline methods are further integrated into open source softwares for the automatic selection of algorithms and hyperparameter values. Auto-WEKA [23] is one of the most popular work towards analyzing machine learning algorithms automatically and setting appropriate hyperparameters in-order to enhance performance. Similarly hyperopt-sklearn is another available software mainly includes random search and TPE for the automatic selection [24]. Inspite of the fact that these tools are automatic, most of them focuses only on accuracy and generalization ability of classifiers, or on the computation cost consisting of testing time [22, 25], while excluding the training time of neural-network based models. Thus existing adaptation tools designed for tuning of hyperparameters need to be further investigated for minimizing the overall response time including both testing and training time.

6.3.3 Gap Analysis

Table 6.1 summarizes the existing approaches with mapping of requirements (suggested in Section–2.4). While classifying the related work, we summarize the gap analysis with limitations as follows:

- *Online Learning of Classifiers*: We consider online training of classifiers as a solution for generalizable multimedia event processing. Existing online learning-based approaches remove the constraints of availability of classifiers by making their decisions on the fly. Due to this reason, these approaches mostly support high maintainability and average to high accuracy due to the use of machine learning models. However, support for low response time and large vocabulary remain not inapplicable in online learning methods.
- *Self-Tuning of Classifiers*: These approaches include the optimization of machine learning models using hyperparameter tuning and self-adaptation of classifiers. Existing work in this category takes the responsibility of delivering high accuracy with the provision of adaptation for optimization. Nevertheless, reducing the training time and supporting a large vocabulary is out of their scope.

6.4 Proposed Approach

The proposed online classifier learning based multimedia event processing model utilizes the publish-subscribe paradigm and leverages neural network-based object detection methods to meet the requirements of dynamic subscriptions. The publish/subscribe system facilitates the smooth interactions between subscribers and publishers sending

multimedia events. The adaptive multimedia event processing engine allows users to subscribe to classes belonging to any domain, monitor multimedia events, and process them using an event-based matching engine, adaptation model, and external resources (shown in Fig. 6.4). The event-based matching engine is responsible for detecting conditions that hold in image events according to the user query. Deep convolutional network-based object detection models are included for the processing of multimedia events with high performance. They are currently being placed in resources that can be changed/adapted on need by the administrator only. The adaptation model has been incorporated for the online configuration of classifiers so that the system can adapt and train new classifiers based on suggested strategies on the arrival of unknown/new subscriptions. The proposed adaptation model derives the best configuration for the considered strategy by analyzing the response time-accuracy trade-off of image processing models (presently object detection). It is important to note that we are using the term "adaptation" for the tuning of hyperparameters based on strategies categorized by response-time. However, other types of adaptations could be incorporated in the future for the enhanced efficiency of the proposed architecture.

The adaptation model has been designed using IBM MAPE-K architecture [172], having *Monitor*, *Analyse*, *Plan*, *Execute* phase, a shared *Knowledge Base*, and managed *resources*, shown in Fig. 6.4. The monitoring function is responsible for receiving subscriptions (in the form of keywords), image events, and other specified requirements. The analyze phase process images and take decisions of training or testing based on the performance of available classifiers. The planning phase configures tunable parameters (assuming Hypothesis-II) to start training based on a decision of the construction of classifiers. The execution phase initiates the training of the classifier. Meanwhile, almost all phases interact with the knowledge base to access configurations, policies, strategies, etc. The last layer contains managed resources (hardware or software) that assist the adaptation engine and presently include training database and neural-network-based models.

6.5 Designing and Implementation

6.5.1 Adaptive Hyper-Parameter based Multimedia Event Processing Engine

The proposed approach for the adaptive multimedia event processing (shown in Fig. 6.4) illustrates the online learning of classifiers on demand along with hyperparameter tuning

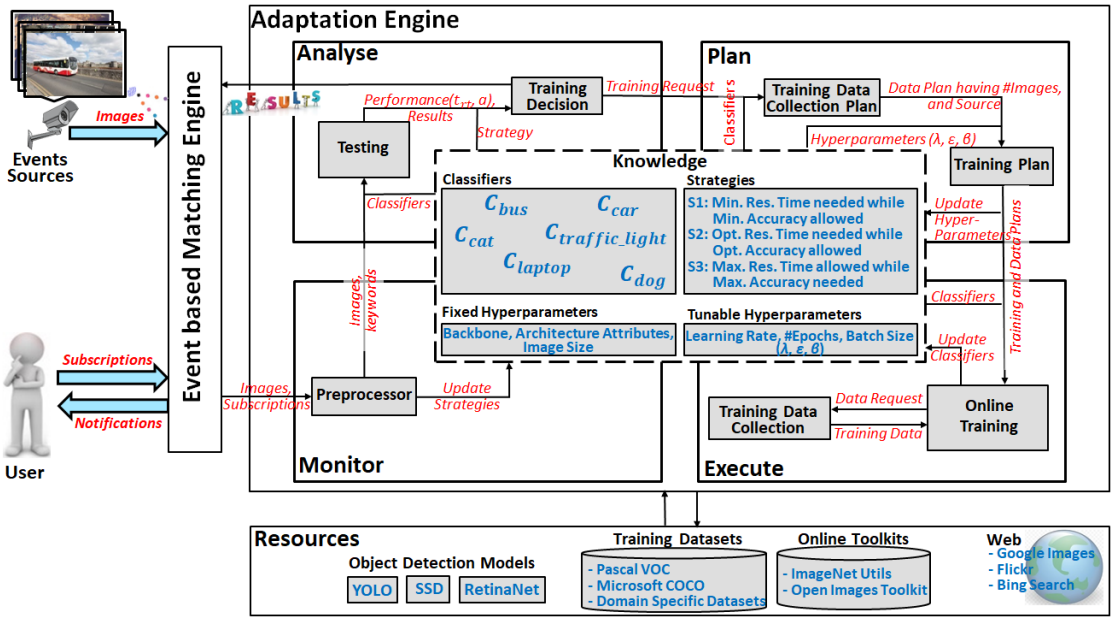


FIGURE 6.4: Adaptation Model for Multimedia Event Processing

based optimisation and addresses the monitoring, analysing, planning, and execution phase using knowledge base and resources as follows:

Monitoring: Firstly the monitor phase is responsible for receiving subscriptions (in the form of keywords like *pedestrian*, *bus*, *cat*, etc.), image events, and any other specific requirement (for instance, *strategies*) suggested by the user. Choice of strategies may vary with applications; presently, we consider mainly three strategies: "Minimum Response Time needed while Minimum Accuracy allowed", "Optimal Response Time needed while Optimal Accuracy allowed", and "Maximum Response Time allowed while Maximum Accuracy needed".

Pre-Processing: It is the responsibility of the preprocessor to update strategies if instructed by the user and then communicate with the analysis phase for providing subscriptions (keywords) and images.

Analyse: The analyse function is designed to evaluate image events and analyse the performance to determine if some changes (specifically training) need to be done. It mainly includes testing and training decision phases, utilise the knowledge base for existing classifiers, and requested strategy.

Testing: The "testing module" processes image events using classifiers belonging to keywords and object detection models with testing configuration parameters from the shared knowledge base and resources.

Training Decision: The training decision phase utilises the results generated by the testing phase and strategies from the knowledge base to start the training or continue testing. It analyses the performance (response-time and accuracy) of the testing module and requests for a training plan accordingly. If response-time (t_{rt}) and accuracy (a) satisfy the requirements of subscribers, the analyse module communicates results to the event matching engine.

Plan: This phase creates or selects a procedure for the training data collection and generates the training plans using classifiers and hyperparameters present in the knowledge base.

Training Data Collection Plan: The data collection plan gets initialised from the training request of the analyse phase; then, it considers available classifiers and training data present in resources to make the data collection plan. It may also consider collecting training data from an external source like automatic data collection tools like OIDv4_ToolKit [28] in the present case. ImageNet-Utils [67] is another typical example of an online data collection tool, which includes more than 1000 categories for user subscriptions. Other than these resources, classifiers for such unknown subscriptions could also be constructed using search engines like *Google Images*, *Flickr*, *Bing Image Search API*, *etc.*, and automatically downloading images using class names.

Training Plan: The training plan receives the data plan with details of sources and the number of images and fetches existing hyperparameters from the knowledge base. It decides the training time by considering the requested strategy and data plan and estimates hyperparameters to give the best performance in the limited response time. Lastly, the training plan updates the hyperparameters (like Learning Rate, Number of Epochs, Batch Size) present in the knowledge base for the proof of Hypothesis-II and invokes the *training* module.

Execute: Execution function provides the environment of online training of classifiers and performs the required changes to update the classifiers necessary for the adaptation of the system.

Online Training: This module mainly performs the training of classifiers for unknown subscriptions using the training plan generated in the previous phase. It may also collect the classifier (if it exists) from the knowledge base and train further following the training plan. Also, training can take place in parallel to testing in the distributed systems. Finally, classifiers get updated in the knowledge base after reaching the training time decided by the planning phase according to the requested strategies.

Training Data Collection: The training module could also instantiate the data collection function to collect training data from an external source (presently OIDv4_ToolKit [28]).

It may also consider details like the number of images, size, quality, etc., from previously generated data collection plans. It provides the requested data within the specified time directed by the online training module. Moreover, it updates resources with collected data for the processing of the same subscription in the future.

Knowledge: It represents a shared knowledge accessible from all phases in different situations and may consist of the following components:

- Classifiers: Testing and training module interacts with classifiers using subscriptions and updates them on need. In the present scenario, classifiers get trained online for the new/unknown subscriptions while collecting training data either online or offline. For instance, for the classes present in Pascal VOC, Microsoft COCO, ImageNet object detection datasets, etc., the model directly collects data from resources offline to train classifiers. However, in the case of a completely new subscription, the model chooses to collect training data online either from existing online data collection toolkits or from web sources.

- Configuration Parameters (Fixed and Tunable): Classifier configuration may vary with the adaptation of tunable hyperparameters. In the current implementation, we fixed the architecture, image size, backbone, etc., and tuned the batch size, learning rate, and the number of training epochs. We use the Tree-structured Parzen Estimators (TPE) method for the tuning of hyperparameters [241], which is the most recent and fixes the limitations of conventional optimisation techniques (please refer to Section–6.3.2).

  Hyperparameters are those parameters of the model whose values get set before the training starts. Setting hyperparameters is critical as they directly affect the behavior of the training and significantly improve the performance (response-time and accuracy) of the model. On the other hand, there is very little research related to ways of choosing hyperparameters for tuning [266]. However, hyperparameters generally classify as *optimiser* hyperparameters and *model-specific* hyperparameters. The optimiser hyperparameters are more focused on the optimisation of the training in terms of efficiency and accuracy. The model-specific hyperparameters are related to the design of the model. A typical set of optimisation hyperparameters for neural networks based models includes learning rate, batch size, and the number of epochs, which we also consider in our adaptation model for the tuning. The learning rate is the most essential hyperparameter that has to be tuned [267]. If the learning rate is too low, it will increase the response time, or if it is too large, the model will never converge. Batch size is also responsible for speed and number of iterations in training. Moreover, larger batch size consumes more memory while smaller batch-size induces noise. Choosing the batch size determines the number

TABLE 6.2: Performance of Existing Object Detection Models

Current Object Detection Models	**Performance**	
	Precision*	**Testing Time (in ms)**
Faster R-CNN [148]	0.27	420.00
R-FCN [268]	0.32	170.00
SSD300 [36]	0.25	21.74
SSD512 [36]	0.29	52.63
DSSD321 [269]	0.28	105.26
DSSD513 [269]	0.33	181.82
YOLOv2 [149]	0.22	25.00
YOLOv3 [166]	0.28	22.00
RetinaNet [37]	0.41	198.00

*average precision on coco test-dev @IOU[0.5, 0.95] [27]

of iterations, and the length of the epoch depends on the number of iterations. Thus the batch size and the number of epochs are directly related to the training time of the model, and we must need to consider such hyperparameters for tuning.

On the other side, attributes that control the architecture of the neural network like the number of layers, activation function, backbone, also fall under the category of hyperparameters, but these parameters are model-specific. We are keeping these elements of specific architectures of each model fixed. However, we are changing the full architectures by changing the object detection models (YOLO, SSD, and RetinaNet) and discussed evaluations in Section–6.6.3.2. Specifically, the SSD model uses *VGG-16* as the backbone and adds 6 convolutional layers while using Softmax as an activation function [36]. RetinaNet model comprises *ResNet-FPN* backbone, a classification subnet, and a box regression subnet, where both classification and box subnet consist of 5 convolutional layers and ReLU based activations [37]. YOLO uses its backbone *darknet* with 24 convolutional followed by 2 fully connected layers and uses linear activation function [35]. The recommended image size for the SSD model is 300×300, RetinaNet model is 800×1333, and YOLO is 448×448. If we modify the image-size, then that would considerably change the testing time with less change in training time. However, this would compromise the accuracy, and we will eventually need more images during training to reach the same accuracy. This will result in more training time and more data collection time in the worst case. Similarly, tuning individually, these parameters could change the specific architectures of object detection models and may enhance the speed of training but not significantly. However, there exist comprehensive reviews [22, 47, 270] that are changing feature extractors (backbones), activation functions, proposals, layers, image size etc. of these models in the field of object detection, but ideal architecture with its parameters is inconclusive to date. Our

model is also extensible for the adaptation within the object detection models by using such existing recommendations and incorporating them in the knowledge base and planning phase of our model. Although, the complexity of our proposed adaptation model will increase with an increase in the number of dimensions, and we will need to give priorities only to a few *model-specific* parameters or only to *optimisation* parameters in the end.

- Strategies: It is important to note here strategies refer to user requirements for performance. Suppose a user permits low accuracy results, but in minimum possible time, then this strategy can be attributed to as "Minimum Response Time needed while Minimum Accuracy allowed"; Conversely, if a user necessitates high accuracy results with no restriction on response time, then one can specify strategy "Maximum Response Time allowed while Maximum Accuracy needed". Similarly, any other choice of response time that supports accuracy between low to high may fall into the category "Optimal Response Time needed while Optimal Accuracy allowed". Please note that *optimal* response time refers to the average response time, which is considered as 60 min in the present model. Its value will always be between the minimum and maximum response time of different applications. The average response time will highly likely provide average accuracy between the model's lowest and highest possible accuracy and be referred to as *optimal* accuracy.

Resources: This component consists of existing image processing models and training datasets. For the demonstration of the proposed model, we are using YOLO, SSD, and RetinaNet for object detection [35–37]; and Pascal VOC and OID [26, 28] with its online toolkit[1] for training datasets. However, resources may include toolkits like ImageNet_Utils[2] or web sources like Bing Scrapper[3], Google Images Downloader[4], Flickr_Photos[5], etc.

Moreover, we could also incorporate other recent object detection models in the future. Presently we analyse the most recent object detection models (shown in Table 6.2): *Faster-RCNN* [148], *Region-based Fully Convolutional Networks (R-FCN)* [268], *Single Shot MultiBox Detectors (SSD)* [36], *Deconvolutional Single Shot Detectors (DSSD)* [269], *You only look once(YOLO)* [35], and *RetinaNet* [37], based on their performance. Here, we focus on the testing time of these models, as after the training of any unknown subscription, the response-time of our model will depend only on the testing-time. Thus

[1]https://github.com/EscVM/OIDv4_ToolKit
[2]https://github.com/tzutalin/ImageNet_Utils
[3]https://github.com/funpokes/bing-image-search
[4]https://github.com/hardikvasa/google-images-download
[5]https://www.flickr.com/services/api/

we chose the YOLOv3 and SSD300 due to their lowest testing time. Moreover, RetinaNet is the most recent among these models, and it is getting popular due to its highest accuracy (to date); we consider it in our experiments. Nonetheless, we could include other object detectors depending on the requested strategies. Please note that the average precision and inference time of object detection models are the best results reported by these models, which may differ in the future with an increase in resources.

6.5.2 Adaptive Hyper-Parameter based Multimedia Event Processing Algorithms

The implementation procedures for online multimedia event processing engine with adaptation are shown in Algorithm 3 and 4, where S_s represents sets of Subscriptions, k: keywords, s: subscribers, S_{IE}: stream of image events, M: object detection model, $\vec{\lambda}$: domain for learning rate, $\vec{\mathcal{E}}$: domain for number of epochs, $\vec{\beta}$: domain for batch size, C_k: classifier for keyword k, and St: strategies respectively.

Algorithm 3 : Adaptive Multimedia Event Processing Engine

Input: *Sets of Subscriptions*(S_s) :
$s_1 : \{\{a\}, \{k_{1_1}, k_{2_1}, ..., k_{j_1}\}\}$, $s_2 : \{\{a\}, \{k_{1_2}, k_{2_2}, ..., k_{j_2}\}\}, ...$
$s_i : \{\{a\}, \{k_{1_i}, k_{2_i}, ..., k_{j_i}\}\}$,
St: Strategy for permissible response time, and
S_{IE}: Stream of Image Events.
Output: Notifications

```
1:  while true do
2:    IE ← Image_Event(S_IE)
3:    m ← count_subscribers(S_s)
4:    for i = 1 to m do
5:      t_a ← {a| a ∈ s_i}
6:      n ← count_keywords(s_i)
7:      for j = 1 to n  do
8:        keyword ← {k_j| k_j ∈ s_i}
9:        objects ← adaptation_engine(IE, keyword, St, t_a)
10:       if (keyword ∈ objects) then
11:         notify(s_i)
12:       end if
13:     end for
14:   end for
15: end while
```

Algorithm 3 gets instantiated with subscriptions consisting of keywords subscribed by multiple subscribers. It also allows subscribers to specify strategies for the permissible response time. Moreover, it continuously monitors the stream of image events while keeping track of the number of subscribers to detect objects according to keywords subscribed by subscribers. Each iteration begins with the arrival time of subscription

Algorithm 4 : Hyperparameter based Adaptation Engine

Input: Image_Event (IE), Keyword (k), Strategy (St), Arrival Time (t_a)
Output: Objects

1: $(C_k, M, \vec{\lambda}, \vec{\mathcal{E}}, \vec{\beta}) \leftarrow identify\_model(St, k)$
2: **if** $C_k \neq \phi$ **then**
3: $(objects, processing\_time, accuracy) \leftarrow process\_image(IE, C_k, M)$
4: **if** $satisfy\_strategy(processing\_time, accuracy, St)$ **then**
5: **return** $objects$
6: **end if**
7: **else**
8: **if** $need\_training\_data(St, k)$ **then**
9: $collect\_data(\#images)$
10: **end if**
11: **end if**
12: $t_t \leftarrow set\_training\_time(St)$
13: $C_k \leftarrow adaptive\_training(k, St, t_t, t_a, C_k, M, \vec{\lambda}, \vec{\mathcal{E}}, \vec{\beta})$
14: **goto** Step 2

and identification of all keywords belonging to subscription. Then for each keyword, we predict objects using our adaptation engine driven by image events, specified strategies, and properties of subscriptions. Finally, subscribers get notified based on identified objects.

The primary role of the adaptation engine (Algorithm 4) is to identify the suitable classifier and predict objects based on specified strategy and subscribed keywords while limiting the processing time up to the permissible response time. First, it attempts to identify the suitable object detection model with specific classifiers suitable for the keyword, along with domain for hyperparameters (λ, $\mathcal{E}$, β). In case $C_k = \phi$, the procedure seeks to find the availability of training data for the keyword in existing object detection datasets present in resources of the model (please see Fig. 6.4). Then we use the training data to train the model for the intended classifier while setting the training time and utilising the derived parameters. Adaptive training, train classifier C_k for time $t_t - t_a$ for keyword k using model M with hyperparameter $(\vec{\lambda}, \vec{\mathcal{E}}, \vec{\beta})$ values mentioned in lookup Table 6.3. Please note I derived these hyperparameter values of different object detection models in evaluation Section–6.6.

Finally, after the training of the classifier, we try to process image events and return objects if processing time (including training and testing), as well as accuracy, is according to the strategy. However, in the worst case, if we do not find the intended keyword-based training data in resources, we also provided the facility of collecting iconic images from the web for such unseen keywords.

It is worth noting that the proposed model is simulated only for adaptation with hyperparameter tuning, but the presented architecture is flexible to incorporate any other

TABLE 6.3: Hyperparameter values for Adaptive Training

Strategy (St)	Model (M)	Batch Size $\vec{\beta}$	Learning Rate $\vec{\lambda}$	#Epochs $\vec{\mathcal{E}}$
S1	YOLOv3	64	0.005315	2
	SSD300	8	0.002612	2
	RetinaNet	1	0.000195	5
S2	YOLOv3	64	0.007935	9
	SSD300	4	0.003600	12
	RetinaNet	2	0.000224	9
S3	YOLOv3	64	0.001	300
	SSD300	32	0.001	120
	RetinaNet	1	1e-5	50

Please see Section–6.6 for Hyperparameter values derivation.

types of adaptation techniques (like domain adaptation) in the future.

6.6 Evaluation

This section first describes the evaluation methodologies, including details of experiment setup, evaluation metrics, and response-time focused strategies. We also show the trade-off of performance with response-time before and after adaptation, along with derived configuration parameters and the experimental results for the proposed strategies. Experiments have been conducted on Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0-26-generic x86_64), with NVIDIA TITAN Xp GPU.

6.6.1 Evaluation Methodology

The evaluations present in this work divides into two categories: online classifier construction with adaptation model and without adaptation model. To test the Hypothesis-II, first we analyse the trade-off between response time and performance (mAP) using default hyper-parameters (without adaptation) on object detection models YOLO, SSD, and RetinaNet [35–37]. Then we change the configuration with hyperparameter tuning to adapt the object detection models for low response time. We present three strategies: *Minimum Response Time needed while Minimum Accuracy allowed*, *Optimal Response Time needed while Optimal Accuracy allowed*, and *Maximum Response Time allowed while Maximum Accuracy needed*, which are part of the proposed adaptation model (shown in Fig. 6.4). Finally, using the performance-response time trade-offs on derived hyperparameters, we identify the suitable models. Since accuracy is not only the best measure for analysing machine learning-based models, we also show the snapshots of confusion matrices for all strategies on multiple subscriptions.

I utilize the images with bounding box annotations of Pascal VOC and OpenImages (OID) datasets for the training of classifiers. Specifically, the number of training images for the subscriptions *cat*, *dog*, *laptop*, *car*, *bus*, *bicycle*, and *football* classes are 1804, 2204, 5528, 2820, 847, 1108, and 4339. To construct the training events, I analyzed annotations of all training images present in Pascal VOC and OID dataset. If bounding box annotations of image consist of **one or more** objects belongs to the particular class (say cat), then I added that image to the training event of that (say cat) class. I consider the testing data of same classed for the testing events. To construct the testing, I again analyzed the annotations of testing data, and if bounding box annotations of image consist of **any** of the classes (cat, dog, laptop and so on), then I added it to testing events set. The number of testing events for the same classes are 384, 538, 355, 1588, 256, 396, and 413 respectively. Lastly, I use the Hyperopt[6] library with Tree-structured Parzen Estimator (TPE) [258], to derive hyperparameters in the multidimensional space of object detection models.

6.6.1.1 Strategies

Based on accuracy-response time trade-off characteristics, the requirements of high-performance execution method (presently object detection methods) can be achieved using the following three main strategies:

Minimum Response Time needed while Minimum Accuracy allowed: The strategy "Minimum Response Time needed while Minimum Accuracy allowed" includes the computation of accuracy that we can achieve by setting limits to response time until it reaches a certain threshold, which is 15 min (including both training and testing time) in the present work by considering requirements of real-time systems. In experiments, I also show that the accuracy of existing object detection models below 15 min is very low on current GPU resources. Thus, setting a threshold of 15 min for minimum response time is influential in the present case. However, 15 min is still a long time for real-time applications, but this delay will only be for the 1st response time. Once we will have the classifier trained the response time (2nd, 3rd, 4th, or so on) will only include the inference time i.e., 0.01 min in our system (detailed in Chapter–7)

Optimal Response Time needed while Optimal Accuracy allowed: Similarly, this strategy "Optimal Response Time needed while Optimal Accuracy allowed" focuses on achieving the optimal accuracy while allowing response time of few hours (1 hour in

[6]https://github.com/hyperopt/hyperopt

the present case) for the training and testing of neural network-based object detection models.

Maximum Response Time allowed while Maximum Accuracy needed: The "Maximum Response Time allowed while Maximum Accuracy needed" would be able to cover existing scenarios of object detection models where models are allowed to train for the extended number of hours to achieve the maximum accuracy. Since this strategy focuses only on maximizing accuracy, it results in high response time, and thus not feasible for real-time scenarios.

In addition to the strategies considered here, we may also design more strategies in the future based on a higher rate of change, approximately-zero-response time, and constant-accuracy. We have conducted experiments using only the above three strategies, since these are highly distinguishable among themselves in terms of response-time, in analysing the best performance on the detection of multimedia events.

6.6.2 Evaluation Metrics

- **Response Time:** It represents the time difference between the arrival and notification of subscription. Suppose a user subscribe at the time "t_a" for "mirror" and there is no available classifier for the detection of *mirror* in the multimedia event processing system (as shown in Fig. 6.3). Thus the proposed model must need to train a classifier which may require data collection (t_{dc}) prior to the training (t_{tr}), and then testing (t_t) of an image event. Finally multimedia event processing system detect events and propagate notifications to user at time "t_b" according to the registered subscription. We can formally define response time (t_r) as:

$$\begin{aligned} t_r &= t_b - t_a \\ &= t_{dc} + t_{tr} + t_t \end{aligned}$$

  It is important to note that this scenario represents Case-2(b) of Section–2.6.

- **Accuracy:** The accuracy is the ratio of correctly predicted observation to the total observations. It is important to note that by *optimal* accuracy in this work, we mean the best accuracy that can be provided by an object detection model in a specified response-time.

- **Mean Average Precision (mAP):** The mAP is the average of the average precision of all classes. It is computed by calculating AP separately for each class, then average over them. So, the resulting mAP could be moderate, but the model

might be useful for specific classes and bad for other classes. Indeed, mAP is widely considered as a good relative metric and has more agreement for the comparison of old and new methods of object detection. To verify the evaluations of mAP of the proposed adaptation model, we also present individual values of precision-recall in Section–6.6.3.2 (Response-Time Driven Precision-Recall Area Under Curve).

- **Confusion Matrix:** A confusion matrix contains information about actual and predicted classifications done by a classification system [271]. A confusion matrix of binary classification has four different categories: true positives, false positives, true negatives, and false negatives. The actual labels (values) form columns and predicted labels (values) form rows. The basic structure of the confusion matrix is shown in Fig. 6.5(a). Here, TP represents the number of true positives (model predicted positive and class is also present), TN represents the number of true negatives (model predicted negative and class is also absent), FP represents the number of false positives (model predicted positive, but class is absent), FN represents the number of false negatives (model predicted negative, but class is present). We show the confusion matrix for multiple subscriptions at a regular interval of time in our evaluations, to show the exact number of actual and predicted subscriptions. Fig. 6.5(b) represents its general structure.

6.6.3 Experiments and Results

6.6.3.1 Online Classifier Construction before Adaptation

Response Time vs Performance of Object Detection Models before Adaptation Fig. 6.6 represents the performance of the proposed model with response time while training from scratch on the arrival of a new subscription. We observe that all three object detection models (YOLO, SSD, RetinaNet) provide low values with the mean average precision (mAP) in low response-time. The maximum performance for the SSD model reaches up to 0.06, YOLO accomplishes mAP 0.09, and RetinaNet achieves mAP 0.20. Among these different models, SSD performs average at 15 *min* of response-time and worse in 1 *hour* of response-time. However, YOLO performs average in 1 *hour* but not in 15 *min*. RetinaNet provides better than both YOLO and SSD models while having mAP of 0.13 in 15 *min* and 0.20 in 1 *hour* for the training from scratch for new subscriptions. We can also note the SSD performance is increased initially and then decreased. This also validates the key difference presented in the SSD model [36], that SSD does not make random guesses like other detectors at the start of the training process, but it assigns ground truth boundary boxes to default boxes. Although 15 min

Confusion Matrix		Actual	
		Positive (P)	Negative (N)
Predictions	Positive (P′)	True Positive (TP)	False Positive (FP)
	Negative (N′)	False Negative (FN)	True Negative (TN)

(a) Structure of Confusion Matrix

Time	Confusion Matrix		Actual								
			$Subscription_1$		$Subscription_2$		...			$Subscription_i$	
			P	N	P	N			...	P	N
t_1	Predictions	P′	TP	FP	TP	FP			...	TP	FP
		N′	FN	TN	FN	TN			...	FN	TN
t_2		P′	TP	FP	TP	FP			...	TP	FP
		N′	FN	TN	FN	TN			...	FN	TN
⋮			⋮	⋮	⋮	⋮			...		
t_j		P′	TP	FP	TP	FP			...	TP	FP
		N′	FN	TN	FN	TN			...	FN	TN

(b) Confusion Matrix with Time for multiple Subscriptions (Classes)

FIGURE 6.5: General Structure Confusion Matrix for Evaluations

and 1 hour is very less time for the training of classifiers (that require up to days), hence training can be very unstable at early stages for any detector.

Default hyperparameters suggested by object detection models that we used for analysing the performance and time trade-off are present in Table 6.4 with their respective accuracy achieved on new subscriptions while using different strategies. The derived accuracies for both strategies S1 (Minimum Response Time needed while Minimum Accuracy allowed) and S2 (Optimal Response Time needed while Optimal Accuracy allowed) state that none of these models are applicable before adaptation and necessitates further investigation after adaptation. Moreover, we can easily conclude that all models are equally suitable only for strategy S3 (Maximum Response Time allowed while Maximum Accuracy needed) at their default configuration, due to having low accuracies on reduced response timings. Please note all models train from scratch without the use of any pre-trained model. Although for the maximum response time of "S3", we are using fully trained weight files provided by object detection models along with their respective recommended backbones *darknet*, *VGG16*, and *resnet50* [36, 37, 149].

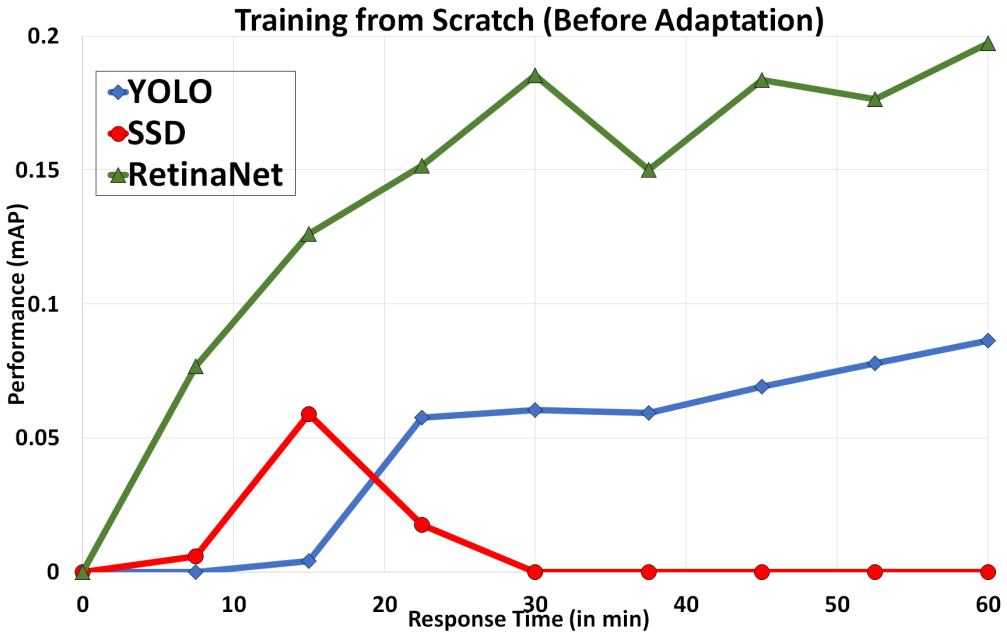


FIGURE 6.6: Performance vs Response Time without Adaptation (for 15-min and 60-min intervals)

Other than accuracy with response time, frame-rate of the object detection models YOLO, SSD, and RetinaNet are 114fps, 21fps, and 7fps respectively. These frame-rates are useful to determine the best model if we have trained classifiers available, where fps represents the number of frames processed per second. The testing time (t_t) given in response time (Section–6.6.2) formulation is the "inverse of fps". On arrival of any "unseen" subscription, the first response time could be 15-min or 1-hour because of the training of the classifier. That subscription (keyword) will become "seen", and the next response time will depend only on the testing time, *i.e.* frame-rate of object detection model. Presently the predicted response-time of the proposed model for *known* subscriptions using YOLOv3, SSD300, and RetinaNet are 0.009, 0.05, and 0.08 seconds respectively.

Results for Proposed Strategies on Selected Object Detection Model Further, experiments have been conducted for strategy S3 using defaults hyperparameter configurations suggested in Table 6.4 on multiple subscriptions. Confusion matrix has been shown by taking SSD as an object detection model presently (can be changed to RetinaNet or YOLO) in Table 6.10, where it contains information about expected and predicted classes detected by the proposed system. Here strategy S3 could serve as an oracle, and its prediction counts show the maximum performance that we could achieve. We can observe that the values of true positives and true negatives are considerably higher than the values of false positives and false negatives for most of the subscriptions

TABLE 6.4: Default Hyperparameters with Accuracy for different strategies.

Object Detection Models	Default Hyperparameters			Accuracy with Response Times		
	Batch Size	Learning Rate	#Epochs	S1: Min Res & Min Acc Time	S2: Opt Res & Opt Acc Time	S3: Max Res & Max Acc Time
YOLOv3	64	0.001	3 10 ~300	0.00%	79.16%	98.53%
SSD300	32	0.001	1 3 ~120	10.08%	54.79%	98.58%
RetinaNet	1	1e-5	4 14 ~50	64.66%	74.87%	98.62%

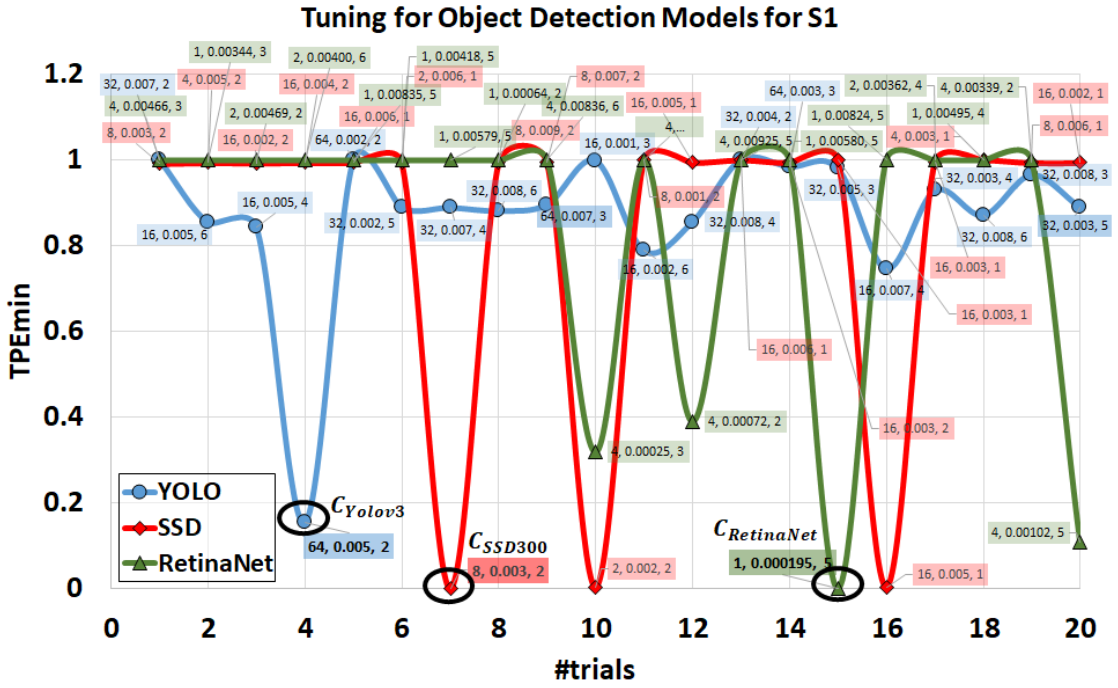
(please see Fig. 6.5 for the details of the confusion matrix). Hence, S3 gives the upper bound of TP and TN, as well as lower bounds of FP and FN. This also concludes that if we allow our model to get trained for the maximum amount of time (up to days), our model will achieve much higher accuracy ($\sim$ 98.58%) even for any previously "unseen" subscription.

However, if a user wants to reduce the first response time, we need to move towards adapting object detection models. Our model achieves this by facilitating strategies S1 and S2 for users and hyperparameter tuning for the adaptation.

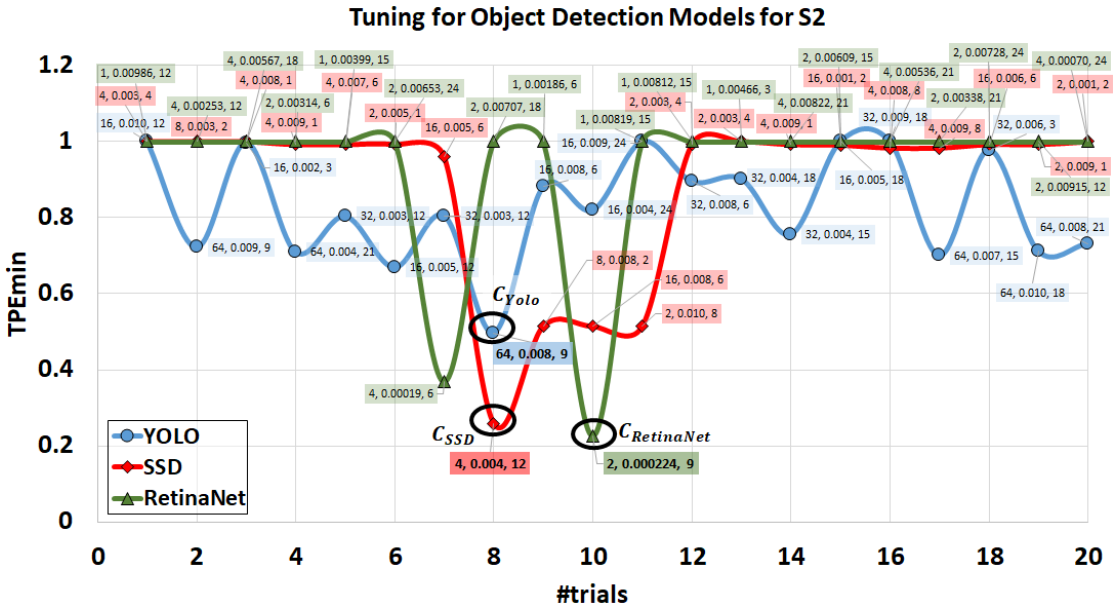
6.6.3.2 Online Classifier Construction after Adaptation

Hyperparameter Tuning Hyperparameter tuning is utilized for the self-optimization of the model on the requested strategies. The goal of tuning is to find the best values of hyperparameters in a given space using a specific function. It mainly requires the objective function to minimise, the space to search hyperparameters, and the method of searching, to output the point of evaluations. Fig. 6.7 represents hyperparameter tuning of object detection models by considering 20 number of trials and the TPE search method [258], which need to be minimized based on mean average precision (mAP). The search space that we used to tune the hyperparameters batch-size (β), learning rate (λ), and the number of epochs ($\mathcal{E}$), is shown in Table 6.5. It is important to note that we chose the domain space of hyperparameters according to the limitation of our resources, and thus could change in the future.

Tuning for the strategy S1 "Minimum Response Time needed while Minimum Accuracy allowed" with specific values of β, λ, and $\mathcal{E}$, for each trial, are shown in Fig. 6.7(a). This attempts to find the parameters that may give the highest performance within 15 min of



(a) Hyperparameter Tuning for Strategy S1: Minimum Response Time needed while Minimum Accuracy allowed, with batch-size, learning rate, and the number of epochs.



(b) Hyperparameter Tuning for Strategy S2: Optimal Response Time needed while Optimal Accuracy allowed, with batch-size, learning rate, and the number of epochs.

FIGURE 6.7: Hyperparameter Tuning for 15-min and 1-hour training

TABLE 6.5: Space defined for Hyperparameter Tuning for the Scratch Training of 15-min and 1-hour.

Object Detection Models	**15-min Training**			**1-hour Training**		
	Batch Size	Learning Rate	Epochs	Batch Size	Learning Rate	Epochs
YOLOv3	{1, 2, 4, 8, 16, 32, 64}	[0.001, 0.1]	[1,6]	{1, 2, 4, 8, 16, 32, 64}	[0.001, 0.1]	[1,24]
SSD300	{1, 2, 4, 8, 16}	[0.001, 0.1]	[1,2]	{1, 2, 4, 8, 16}	[0.001, 0.1]	[1,12]
RetinaNet	{1, 2, 4}	[0.00001, 0.01]	[1, 6]	{1, 2, 4}	[0.00001, 0.01]	[1, 24]

training for any "unseen" subscriptions. As the full training time of each object detection model is up to days, it is hard to train a model within only 15 min (or even in 1 hour). Thus, no model indicates any accuracy for most of the combinations of hyperparameters and shows the maximum value for TPE (which is based on the inverse of the mAP). However, we find a few combinations of hyperparameters that give average accuracy even within the 15 minutes of training time, and that shows the sudden minimum for those few values. In the case of YOLO, we observe the model is reaching a minimum TPE for the largest batch-size of 64. Moreover, it also requires a higher learning rate (0.005) close to the highest value (0.008) in the case of YOLO. Although the number of epochs found is 2 for the minimum TPE, the highest value of the number of epochs we could achieve in 15 min is 6.

We found that the SSD model is slowest in training and cannot train more than 2 epochs in 15 min. In this case, we get the minima at three points: (8, 0.003, 2), (2, 0.002, 2), and (16, 0.005, 1), which proves that even with the lower number of epochs, we can achieve average accuracy by altering the batch-size and keeping high learning rates. We choose $\beta = 8$, $\lambda = 0.003$, and $\mathcal{E} = 2$ for SSD in our experiments, which could switch to any other two data points. RetinaNet model achieves its minimum value at data point (1, 0.000195,5), which represents the lowest learning rate as well as the smallest batch size among all trials. However, the RetinaNet model reaches up to 5 epochs with such low learning rates within 15 min of training.

Similarly, tuning for finding the best parameters for strategy S2 "Optimal Response Time needed while Optimal Accuracy allowed", is shown in Fig. 6.7(b). Here we found the minimum value of TPE function for YOLO model at data point (64, 0.008, 9) which again (same as S1) we found at the highest value of batch-size, and high learning rate, while having a low number of epochs (9) as the highest value achieved could be 21. The SSD model found its minima at (4, 004, 12), which indicates the highest number of epochs in 1-hour training. RetinaNet for S2 follows the same trend as S1 and found

its minimum point at the lowest learning rate (0.000224), low batch-size (2), and epoch value reached till 9 where possible highest value of epoch could be 24.

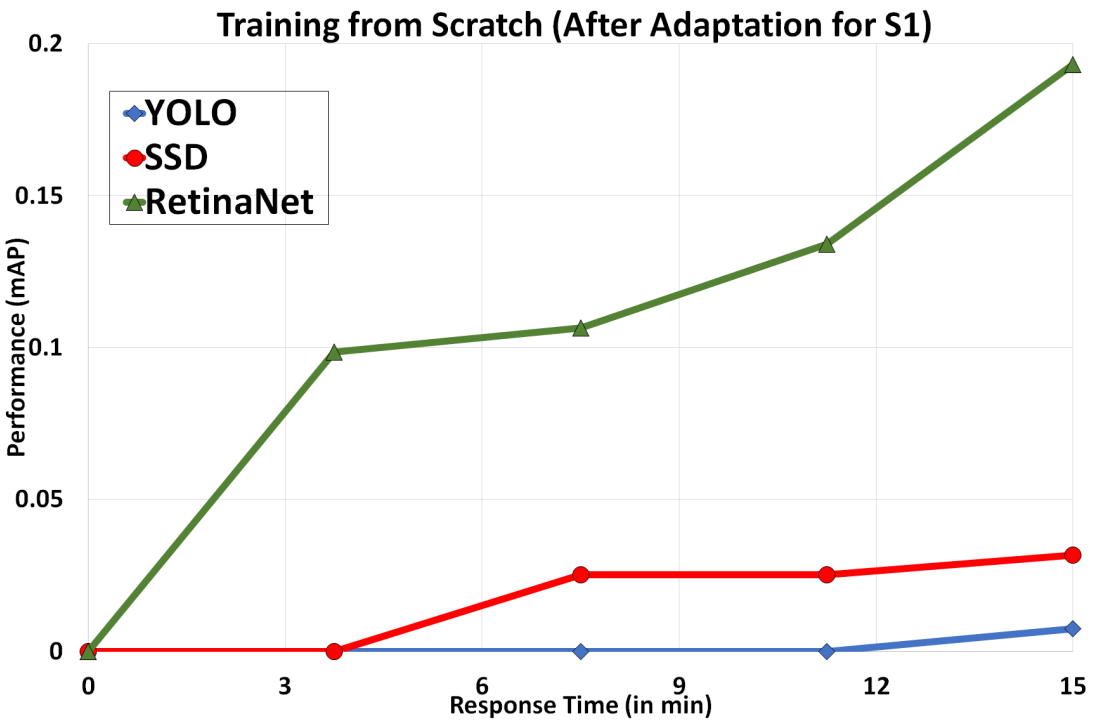
We use the derived data points to investigate the maximum performance we can achieve using different models while analysing the trade-off of performance with response time for strategies S1 and S2.

Response Time vs Performance of Object Detection Models after Adaptation Fig. 6.8 represents a trade-off of performance (mAP) with response time after adaptation (tuning hyperparameters) of the proposed model to process new subscriptions with strategies S1 and S2. The performance of the proposed multimedia event detection model has been evaluated on the best configuration hyperparameters (*learning rate*, *batch size*, and the *number of epochs*) derived in previous Section 6.6.3.2, for the training of 15 min and 1 hour. We observe that the RetinaNet model performs better than YOLO and SSD for strategy S1 (please see Fig.6.8(a)). Moreover, its performance is also enhanced to the precision of 0.20 (after adaptation) from 0.13 (before adaptation). Similarly, Fig. 6.8(b) shows the mAP for strategy S2 with a response-time of 1-hour. Here, also RetinaNet outperforms, and its precision increased from 0.20 (before adaptation) to 0.32 (after adaptation).

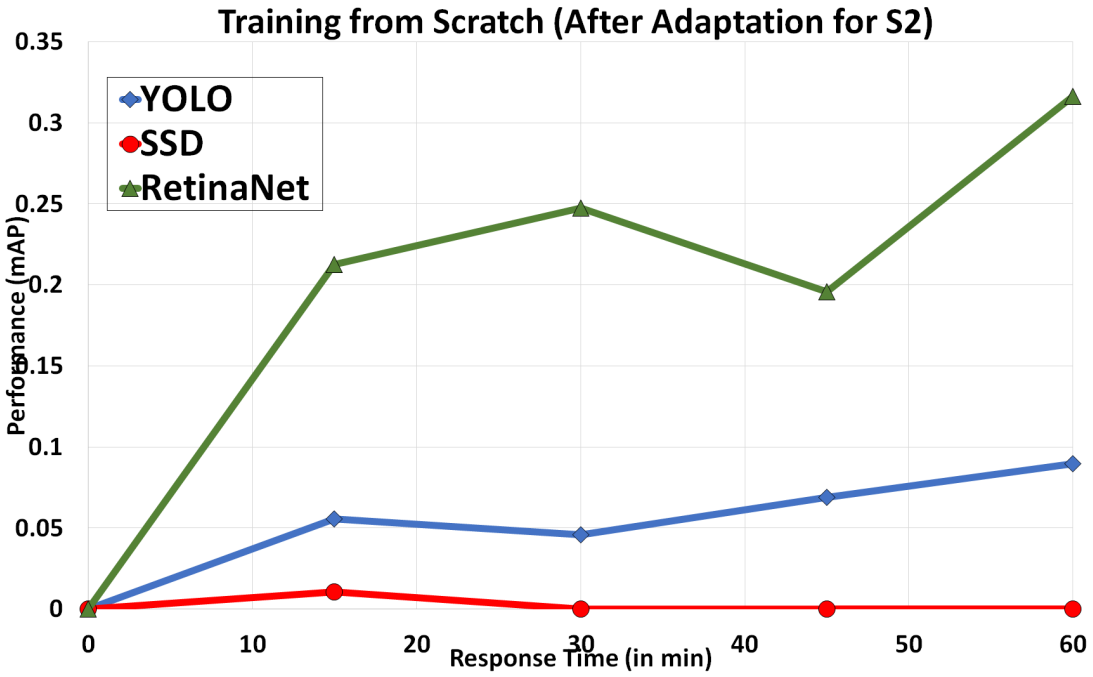
Results of achieved accuracies after adaptation, along with derived hyperparameters of object detection models for both strategies S1 and S2, are shown in Tables 6.6 and 6.7. We found that the accuracy of each model before adaptation (Table 6.4) increases after adaptation (Table 6.6), for strategy S1. Specifically it increases from 0.00% to 5.66%, 10.08% to 47.32%, and 64.66% to 79.00% for YOLO, SSD, and RetinaNet respectively. Correspondingly, we also get better accuracy for strategy S2 after adaptation (Table 6.7) than before adaptation (Table 6.6). YOLO increased from 79.16% to 82.82%, SSD slightly changed from 54.79% to 54.81%, and RetinaNet considerably increased from 74.87% to 84.28%.

We conclude that RetinaNet is performing best among all object detection models on such low training times. Thus, we can easily consider RetinaNet with its derived configuration to detect objects for both strategies S1 and S2. Moreover, the enhancement in performance on such low training time (15 min and 60 min) of object detection models on the tuning of hyperparameters (i.e., after adaptation) validates Hypothesis-II.

Since recall is also a popular evaluation metric but not regarded as useful for comparing object detection models, we present an analysis of precision-recall with the change in response-time in the next section. Other than communicating the change in values of recall with response-time, these precision-recall curves clearly show that the Area Under



(a) 15-min Training for Strategy: Min. Res. Time needed while Min. Acc. Allowed



(b) 60-min Training for Strategy: Opt. Res. Time needed while Opt. Acc. allowed

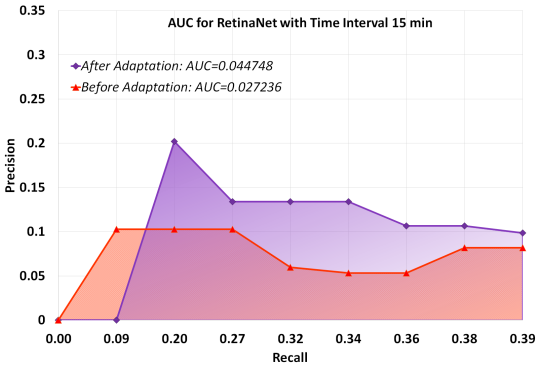
FIGURE 6.8: Performance vs Response Time after Adaptation (for 15-min and 60-min intervals)

Curve (AUC) is relatively bigger after adaptation than before adaptation. Higher values for time-based AUC for RetinaNet also support its high precision and recall for 15 min and 1-hour training.

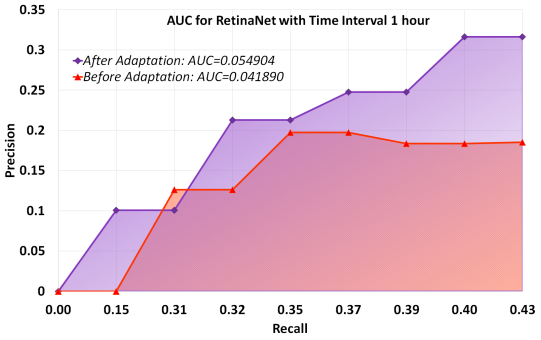
Response-Time Driven Precision-Recall Area Under Curve (AUC) The precision-recall curves visualise the performance of classification models while summarizing the trade-off between precision and recall using a range of thresholds. A high area under the curve represents high scores for both precision and recall, which shows that the classifier is returning accurate results (high precision) and the majority of all positive results (high recall). Area Under Curve (AUC) is an approximation of the area under the precision-recall curve [272]. AUC is desirable to evaluate which model is performing better and what should be the value of the threshold to achieve maximum precision as well as recall. However, in our case, we already have values of threshold evaluated by different object detection models (YOLO:0.25, SSD:0.45, RetinaNet:0.50). Moreover, we need to assess these models before and after the proposed adaptation within the short interval of response-time. Thus, we show the precision-recall curves by plotting all data points of precision and recall computed within the response time of 15 min and 1 hour in Fig. 6.9, and verify that AUC values *after adaptation* are relatively bigger (higher) than *before adaptation*.

The performance of RetinaNet before and after adaptation is shown in Fig.6.9(a) and 6.9(b). It can be seen the precision-recall curve covers more area after adaptation in both cases of 15 min and 1-hour response-time, thus also have relatively better values for AUC as compared to the actual AUC of the RetinaNet model (i.e., without/before adaptation) within such short training time. Analysis of the SSD object detection model (Fig. 6.9(c) and 6.9(d)) shows its lower performance and verifies the improvement in performance after adaptation than before. AUC for YOLO after adaptation is better than before adaptation for response-time of 1 hour (Fig. 6.9(f)). Here, the performance of YOLO for the 15 min training time gets decreased after adaptation, which could be the reason YOLO is still struggling with accuracy and not considered very reliable compared to other object detection models [47, 273]. However, other than the AUC of YOLO for 15 min response-time, we can conclude that the proposed adaptation strategy is effective in all cases using all object detection models (Fig. 6.9).

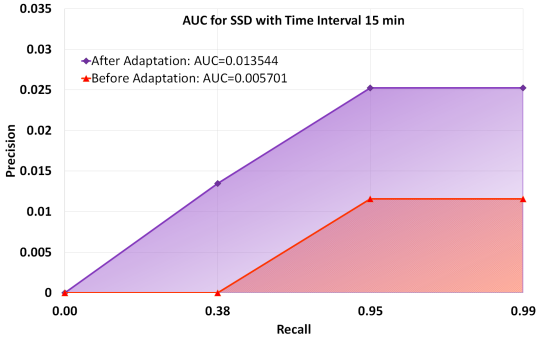
It also verifies that the RetinaNet model with adaptation performs the best for both cases of 15 min and 1 hour response time. Thus RetinaNet with its derived configuration is suitable for both Strategies 1 and 2 shown in Tables 6.6 and 6.7. Specifically, the peak values found for precision and recall within a time interval of 15 min are 0.20 and



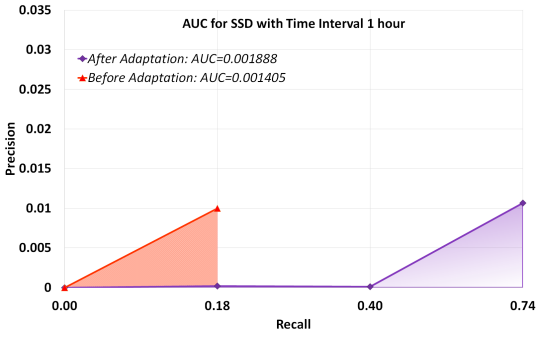
(a) AUC for RetinaNet (within 15-min)



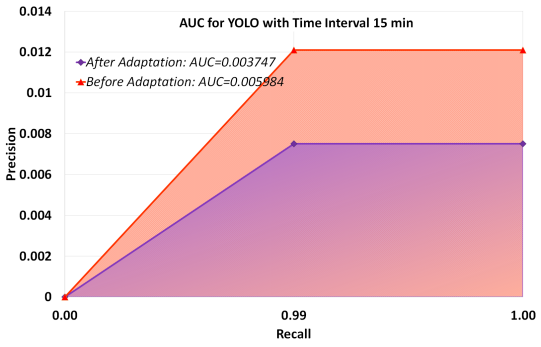
(b) AUC for RetinaNet (within 1-hour)



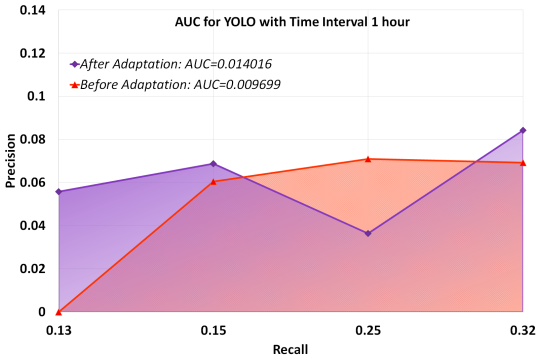
(c) AUC for SSD (within 15-min)



(d) AUC for SSD (within 1-hour)



(e) AUC for YOLO (within 15-min)



(f) AUC for YOLO (within 1-hour)

FIGURE 6.9: Area Under Curve (AUC) Before and After proposed Adaptation within Response-Time interval of 15 min and 1 hour using different Object Detection Models

0.20, respectively. Lastly, the highest values of precision and recall are 0.32 and 0.43, respectively, for the response time of 1 hour.

Results for Proposed Strategies on Selected Object Detection Model Table 6.8 represents the results of the proposed adaptation model for Strategy-1 "Minimum Accuracy and Minimum Response Time" using RetinaNet as an object detection model with 15-min of training from scratch. Similarly, Table 6.9 represents the results for Strategy-2 "Optimal Accuracy and Optimal Response Time" until one hour. Here values of true positives (TP) and true negatives (TN) shown in light colour should increase with

time, and values of false positives (FP) and false negatives (FN) shown in dark colour should decrease with time.

We observe that even within 15 min of training, we get remarkable counts for TP and TN, following high FP and FN values. Table 6.8 shows that TPs are increasing extensively in four cases and decreasing in three cases with time. However, TNs are increasing in three cases, and in four cases, its count is decreasing. Similarly, FP and FN are decreasing considerably in three cases, but not in four cases. So, we conclude that the model is not stable in 15 min of training and requires more time to train completely. Despite that, it provides an average accuracy of 79.00% (using RetinaNet) within 15 min, and we could consider it suitable in situations where we need a quick response and compromise in the accuracy is allowed.

Additionally, when we apply the derived hyperparameters for the strategy S2, TP values increase for most of the classes within 1-hour training compared to values at 15 min of training. Here, values of TNs are increasing and decreasing with time, and values of FPs are decreasing and increasing as well. Although FNs are decreasing in the majority of cases for 1 hour of training. The average accuracy computed from the confusion matrix (shown in Table 6.9) is 84.28% for S2 using RetinaNet.

It is worth noting that the total number of input images at different time intervals is the same in each subscription, and the number of instances detected in an image could be different for distinct models. For instance, if we give an input image consisting of two cats, and our model after 7 min of training detects five cats, then we will have $TP = 2$, $FP = 3$, $TN = 0$, and $FN = 0$ (*i.e.*, the *total number of instances* $= 5$ at 7 min). On the other hand, if our model after 15 min of training detects three cats, we will have $TP = 2$, $FP = 1$, $TN = 0$, and $FN = 0$ (*i.e.*, the *total number of instances* $= 3$ for 15 min). These multiple detections in an image make the resulting total number of instances different in the confusion matrix of *object detection*, unlike the case of conventional *image classification*, where an image could either just "belong" or "not belongs" to a particular class. Nevertheless, there is still a gap between the values (TP, FP, TN, and FN) for strategies (S1 and S2) and oracle strategy S3 (Table 6.10 discussed in Section–6.6.3.1), which is explicit because of their large gap in response-time.

Apart from these experiments, the performance of the proposed model will highly depend on the amount of resources, including its execution environment. Response-time will get reduced further on more GPUs, and accuracy could be enhanced by lowering the learning rates of object detection models. Moreover, the resolution of the input stream of images and their quality will also impact the accuracy of models. However, Model-II is independent of change in application due to constantly training from scratch. I will discuss the limitations of domain adaptation in next Chapter–7.

TABLE 6.6: Derived Hyperparameters with Accuracy for Strategy-1

Object Detection Models	**Computed Hyperparameters**			**Accuracy for S1: Min. Res. & Min. Oct. Time**
	Batch Size	Learning Rate	#Epochs	
YOLOv3	64	0.005315	2	5.66%
SSD300	8	0.002612	2	47.32%
RetinaNet	1	0.000195	5	79.00%

TABLE 6.7: Derived Hyperparameters with Accuracy for Strategy-2

Object Detection Models	**Computed Hyperparameters**			**Accuracy for S2: Opt. Res. & Opt. Acc. Time**
	Batch Size	Learning Rate	#Epochs	
YOLOv3	64	0.007935	9	82.82%
SSD300	4	0.003600	12	54.81%
RetinaNet	2	0.000224	9	84.28%

6.7 Conclusion and Discussion

In this chapter, I removed the limitation of pre-trained classifiers to process unseen concepts in multimedia event processing. While analyzing literature, I demonstrate that the training of classifiers online is a solution for dynamic seen/unseen concepts of smart cities, and online learning approaches are not focused on training time. Similarly, their optimization approaches are suitable for adaptation but not for short response time. I proposed an adaptive approach for multimedia event processing using online classifier construction of object detection models for the handling of unseen subscriptions with a low response-time. The proposed model is optimised with the tuning of hyperparameters of existing object detection models YOLOv3, SSD300, and RetinaNet. Experiments demonstrate that the trade-off between performance and training time with adaptation could be useful to reduce the overall response time by compromising the accuracy. The proposed system achieves an accuracy of 79.00% with 15 min training and 84.28% with 1-hour of training on a single GPU, which is reasonable for the detection of objects for unseen subscriptions on such low training times. The difference in the performance of before and after adaptation of tuning of hyperparameters of proposed model validates Hypothesis II by speeding up the training and enhanced accuracy.

However, one of the limitations of this model is that it cannot adapt among domains having related (semantically/visually) concepts (presently classes) of real-world events. Thus we extend the proposed model in the next Chapter–7 by introducing a domain adaptive classifier construction approach for seen to unseen concept knowledge transfer.

TABLE 6.8: Confusion Matrix for Strategy S1: Minimum Response Time needed while Minimum Accuracy allowed

Time	Matrix		Subscriptions (Expected)													
			Cat		Dog		Laptop		Car		Bus		Bicycle		Football	
			P	N	P	N	P	N	P	N	P	N	P	N	P	N
7 min	Predictions	P'	185	3050	24	652	37	160	112	230	183	496	178	214	0	0
		N'	180	4768	506	4928	162	365	1429	4846	71	479	211	481	220	401
15 min		P'	0	2	301	7489	64	201	219	281	2	19	37	57	82	27
		N'	370	4952	229	4674	135	339	1322	4749	252	495	352	491	138	321

TABLE 6.9: Confusion Matrix for Strategy S2: Optimal Response Time needed while Optimal Accuracy allowed

Time	Matrix		Subscriptions (Expected)													
			Cat		Dog		Laptop		Car		Bus		Bicycle		Football	
			P	N	P	N	P	N	P	N	P	N	P	N	P	N
15 min	Predictions	P'	322	7650	13	218	52	47	246	150	110	127	52	143	69	13
		N'	48	4638	517	4939	147	349	1295	4721	144	485	337	491	151	335
30 min		P'	166	2058	10	68	135	373	144	36	222	847	163	615	150	110
		N'	204	4786	520	4942	64	275	1397	4823	32	478	226	482	70	257
45 min		P'	160	1444	13	78	58	27	628	555	181	203	228	161	117	24
		N'	210	4793	517	4939	141	343	913	4475	73	480	161	477	103	286
60 min		P'	117	724	126	575	127	93	680	555	173	950	133	105	116	25
		N'	253	4835	404	4828	72	278	861	4441	81	480	256	483	104	288

TABLE 6.10: Confusion Matrix for Strategy S3: Maximum Response Time allowed while Maximum Accuracy needed

Time	Matrix		Subscriptions (Expected)													
			Cat		Dog		Laptop		Car		Bus		Bicycle		Football	
			P	N	P	N	P	N	P	N	P	N	P	N	P	N
Upto Days	Predictions	P'	318	52	438	69	97	20	1018	523	180	74	280	109	15	7
		N'	19	4610	92	4475	102	224	79	4141	20	475	23	468	205	190

Chapter 7

Domain Adaptation based Multimedia Event Detection

7.1 Introduction

One of the evident solution of reducing response-time from previous Chapter-6 is to allow adaptation among domains. I formulate this in research Hypothesis-III as "*Domain adaptation based Multimedia Event Detection model relies on the fact that if transferring of knowledge from one domain to another (say $A \rightarrow B$) can improve the performance as compared to fine-tuning of pre-trained models (like $C_{P_{ImageNet \rightarrow B}}$) or training of classifier from scratch (C_B); then there will always be a decrease in response-time with increase in accuracy of constructed classifier ($C_{A \rightarrow B}$) than the classifier trained from pretrained model (like $C_{P_{ImageNet \rightarrow B}}$) or training from scratch (C_B)*". This Chapter-7 mainly tackles the third Research Question 3(a) "*How can we answer multimedia event based queries online consisting of unseen subscriptions (unbounded vocabulary), using domain adaptive classifier construction approach with knowledge transfer from seen subscriptions (bounded vocabulary) while achieving high accuracy and minimizing the response time?*".

In this chapter, I introduce the notion of adaptation among classifiers (either inter or intra domain) for reducing response-time. Transfer learning is well-known for easy knowledge transfers among domains and could help either in switching (transforming) from one classifier to another (like bus $\rightarrow$ car) or in the construction of a completely new classifier (like ball). Presently, adaptive approaches [274, 275] are focused on the generalization ability of classifiers for the enhancement of accuracy. They do not analyze the response time of the process of transfer and its impact on accuracy. Thus, there is

a need to investigate the problem (detail in Section–7.2) of online construction of classifiers which can allow adaptation among domains for seen/unseen concepts considering response time and accuracy.

After the problem formulation, I discuss the background associated with transfer learning and analyzed the gap in existing knowledge transfer approaches in Section–7.3. Next, I propose an adaptive multimedia event processing model presented in Section–7.4 and 7.5 that leverages transfer learning-based techniques for domain adaptation to handle unseen/new subscriptions within an acceptable time frame. The results in Section–7.6 indicate that the proposed framework can achieve accuracy between 95.14% to 98.53% within response time of $\sim 0.01min$ to $\sim 30min$ for seen or completely unseen subscriptions using YOLOv3 object detection model on real-time multimedia events. Lastly I conclude in Section–7.7 with discussion on limitations of requirement of annotated bounding boxes for online training.

7.2 Problem Overview

Multiple applications may require handling of numerous seen/unseen concepts which may belong to the same/different domains with an unbounded vocabulary. Although deep neural network-based techniques are effective for image recognition, the limitation of having to train classifiers for unseen concepts will lead to an increase in the overall response-time for users. Since it is not practical to have all trained classifiers available, it is necessary to address the problem of training of classifiers on demand for unbounded vocabulary with provision of domain adaptation.

7.2.1 Preliminaries

Domain Adaptation can utilize the knowledge of source data distribution to identify different (but related) target data distribution [276]. The model learns from the source domain consisting of labeled data and from the target domain using unlabeled/labeled data, and in most use-cases, data available in the source domain is much more than the target domain [276, 277]. Specifically, domain adaptation is getting popular in the field of deep learning via transfer learning techniques.

Transfer learning is a machine learning technique that reuses the pre-trained model on a new problem, thus responsible for transferring knowledge from one domain to another [274, 278]. For example, a classifier that can detect a *bus*, could be useful in training *car* detector by knowledge transfer. The mathematical representation of transfer learning is given in Section–7.3.

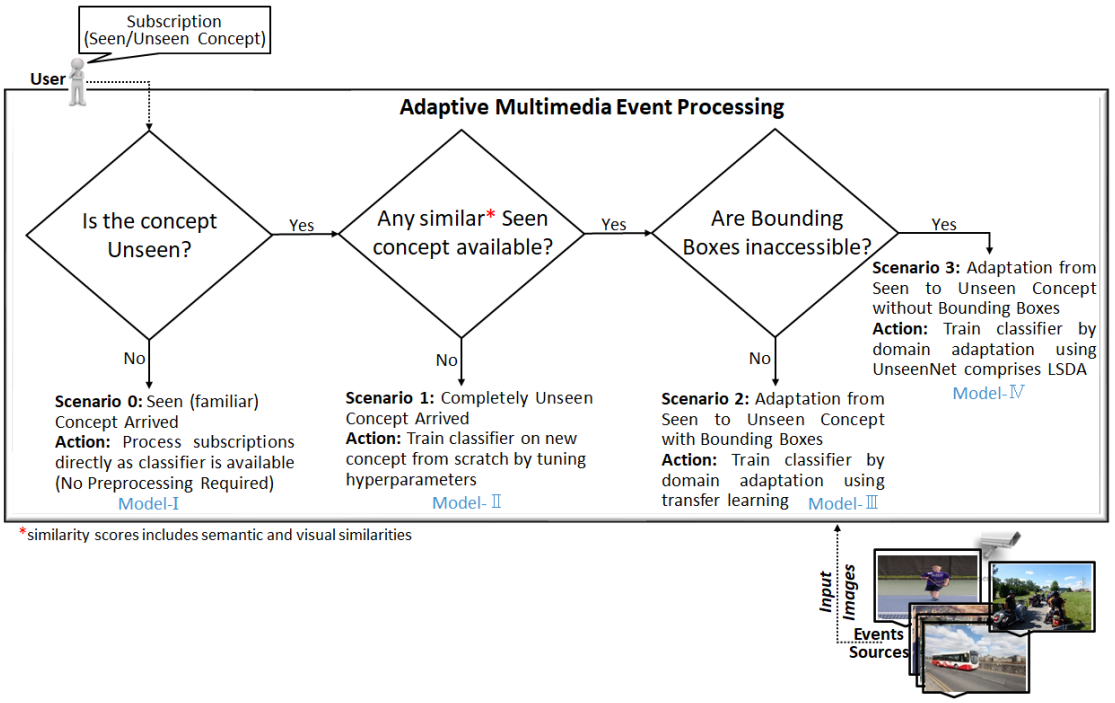


FIGURE 7.1: Scenarios for Multimedia Event Processing adhering to Seen/Unseen Concept Problem

7.2.2 Motivational Scenarios

Consider the baseline scenarios of detecting seen/unseen concepts for analyzing multimedia events, as shown in Fig. 7.1. Other than the scenario of seen and completely unseen concept (detailed in Chapter—5 and 6), the most common scenario is Scenario-2, where we need to process unseen concept, and this new concept is also visually/semantically similar to the seen concept. The specific problem associated with Scenario-2 is described in Fig. 7.2, where we need to reduce the response time for cases where the intended classifier is not available but similar classifiers available. Suppose a user subscribes to the detection of class "car", unseen to the multimedia event processing model. If a model already consists of related classifiers (like *bus* in the present case), we need to train classifiers for such partial unseen concepts in a reduced response time. In that case, we need a provision of domain adaptation in the multimedia event processing model, which does not exist in conventional domain-specific approaches. For instance, with knowledge transfer, we could train a classifier for unseen class "car" by applying transfer learning on seen class "bus" while using annotated bounding boxes based on car class data.

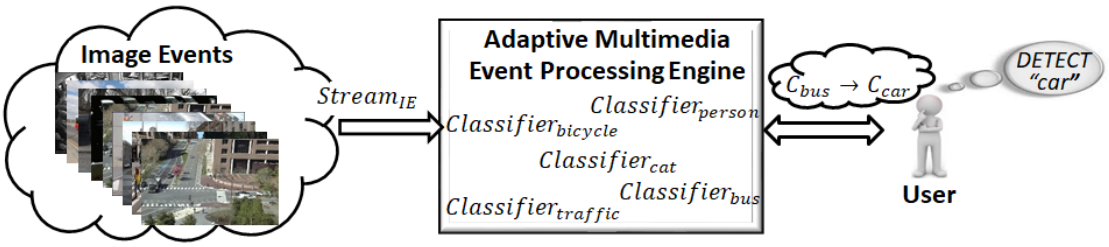


FIGURE 7.2: Scenario-2: Adaptation Possible from Seen to Unseen Concept with Bounding Boxes (Application of Model-III: Domain Adaptation based Multimedia Event Detection)

7.2.3 Problem Statement

We formulate the problem as "How can we answer multimedia event based queries online consisting of unseen subscriptions (unbounded vocabulary), using domain adaptive classifier construction approach with knowledge transfer from seen subscriptions (bounded vocabulary) while achieving high accuracy and minimizing the response time?"

7.3 Background and Related Work

Domain adaptation is the ability to utilize the knowledge of old domains to identify new domains. The model learns from the source domain consisting of labeled data and the target domain using unlabeled/labeled data, and mostly more data is available in the source domain [276, 277]. An example of domain adaptation is shown in below Fig. 7.3.

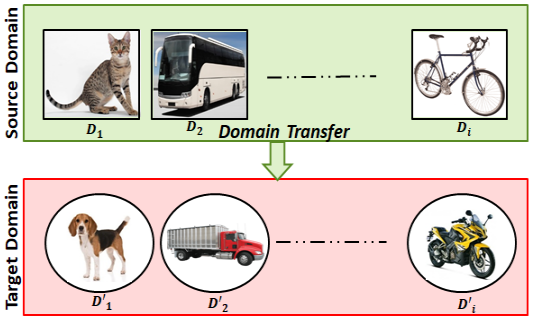


FIGURE 7.3: Domain Adaptation

The term *Transfer learning* is basically used to reuse a pre-trained model on a new problem, thus responsible for the transfer of knowledge from one domain to another. Specifically, domain adaptation via transfer learning in the field of deep learning is getting popular due to its ability to train neural networks with comparatively fewer data [274, 278]. Formally we can define transfer learning with adaptation in the domain as:

Assume we have two domains a source domain $D^s = X^s, P(X^s)$ with $T^s = Y^s, P(Y^s|X^s)$ and a target domain $D^t = X^t, P(X^t)$ with $T^t = Y^t, P(Y^t|X^t)$, where X and Y are random variables, $P(X)$ denotes the marginal probability distribution, and $P(Y|X)$ is the conditional probability distribution. If $D^t \neq D^s$ or $T^t \neq T^s$, but source domain is related to target, then the model trained on (D^s, T^s) can be used to learn $P(Y^t|X^t)$, which is known as transfer learning [276].

Since transfer learning makes machine learning algorithms more efficient, knowledge transfer approaches are essential to include in multimedia event processing systems to adapt domains. We analyze below existing knowledge transfer approaches for the training to testing data domain transfer and expressly object detection domain shift distribution.

7.3.1 Knowledge Transfer from Training to Testing Data Distribution

Domain transfer includes adapting machine learning models acquired knowledge of a particular visual domain, and transfer to new imaging conditions. Many approaches [275, 279–281] with supervised/unsupervised transfer learning have been proposed for domain adaptation and are mainly focused on utilizing the generalization ability for increasing accuracy not the overall response time. Sun et al. [275] proposed an efficient method for unsupervised domain adaptation called CORrelation ALignment (CORAL). Its evaluations on deep and shallow features of object recognition and sentiment analysis confirm the suitability of CORAL for different tasks of computer vision and natural language processing. Another representation learning-based domain adaptation approach proposed by Ganin et al. [279] helps image classification. Like CORAL, this approach also utilized shallow and deep feed-forward architectures and achieved new state-of-the-art results. Long et al. [280] proposed a Deep Adaptation Network (DAN) architecture to generalize models well for the domain adaptation scenarios. DAN attempted to enhance the transferability of features in higher layers of the neural network. Its empirical analysis of A-Distance demonstrates the efficacy over standard domain adaptation benchmarks. Yoshua Bengio [281] also discussed the transfer learning for the different training data distribution problems. The work exploits the utility of unsupervised pre-training of representations. It shows the improvements in error rates for classification and low transfer ratios for stacked denoising autoencoders (SDA) along with demonstrations of the suitability of deep learning with transfer learning.

TABLE 7.1: Analysis of Related-Work with identified Requirements for Knowledge Transfer

Category	Approach	**Requirements**			
		High Accuracy for Multimedia Events	Low System Response Time	Support for Large Vocabulary	Maintain-ability
Knowledge Transfer from Training to Testing Data Distribution	CORAL [275]	Average Accuracy	N.A	N.A	Unsupervised Domain Adaptation
	DAN[280]	High Accuracy	N.A	N.A	Generalize to Domain Adaptation
	DANN [279]	High Accuracy	N.A	N.A	Easy Domain Adaptation
	SDA [281]	High Accuracy	N.A	N.A	Transferable
Knowledge Transfer for Object Detection Domain Shift distribution	Objects and Scene Representa-tions Transfer[282]	High Accuracy	N.A	N.E	Flexible for Domain as well as Target Transfer
	Domain Adaptive Faster R-CNN [60]	Average Accuracy	N.A	N.E	Applicable for Image/ Instance-Level Domain Shift
	Regularized Cross-Domain Transforms [283]	Average Accuracy	N.A	Full Support Theoreti-cally	Adaptable for new Image Conditions

N.A: Not Applicable, N.E: Not Evaluated

7.3.2 Knowledge Transfer for Object Detection Domain Shift Distribution

Event recognition in still images by transferring objects and scene representations has been proposed in work [282], where the correlations of the concepts of object, scene, and events have been investigated. This work proposed techniques to exploit the knowledge from other networks and develop initialization-based, knowledge-based, and data-based transfer techniques. The evaluations of the proposed model on multiple event domains show a reduction in over-fitting and improving generalization ability. Another domain adaptation approach [60] based on the Faster R-CNN object detection model has been

proposed recently to reduce the domain discrepancy and enhance the effectiveness of cross-domain object detection. The approach tackles the image-level shift (like image style, illumination, etc.) and instance-level shift (like object appearance, size, etc.) for the domain shift. The results demonstrate that the proposed approach outperforms baseline Faster R-CNN for different scenarios of the domain transfer. One of the first studies of domain shift in the context of object recognition is presented by Saenko et al. [283]. Here, they introduced a method to adapt object models designed for particular visual domains to new imaging conditions. The approach minimizes the effect of domain-induced changes by learning transformations in a supervised manner. Evaluations prove that model is flexible from moderate to significant changes in the imaging conditions with few or no target domain labels.

7.3.3 Gap Analysis

Table 7.1 summarizes the existing approaches with mapping of requirements (suggested in Section–2.4). While classifying the related work, we summarize the gap analysis with limitations as follows:

- *Knowledge Transfer from Training to Testing Data Distribution*: Existing knowledge transfer approaches focused on differences in training and testing data distribution are mostly generalizable and achieve high accuracy. However, the training time is not the issue in these approaches and does not explicitly focus on adapting individual object categories.
- *Knowledge Transfer for Object Detection Domain Shift Distribution*: In this case, domain shift represents different changes in view-points, weather conditions, backgrounds, image quality, style, sketches, image size, etc. Such transfer learning approaches are popular because of their less need for training data to provide accurate results. Besides these characteristics, domain transfer approaches should also focus on their ability to train models in less training time. It is worth noting that one of the existing techniques can also support a large vocabulary theoretically and require more focus for their practical applications.

7.4 Proposed Approach

The proposed approach of processing multimedia events is based on the construction of classifiers on a need-to-know basis to answer subscriptions that are previously seen/unseen by leveraging the transfer learning-based domain adaptation techniques and deep

convolutional neural network-based object detection models to event processing systems. Subscriptions are expressed using a "Detect" operator with a keyword (*bus*, *car*, *ball*, *person* etc.) for the object, with the same user interface for publish/subscribe services as described in the work of Chapter-4 based on multimedia event processing for IoMT.

On each new arrival of subscription, the proposed model first identifies if any similar classifier is available or any possibility for domain adaptation. Second, it performs the training of classifiers on the need for the intended new subscription. However, domain adaptation from a pre-trained model or the related classifier may require different transfer learning techniques that we analyze for Hypothesis-III. More specifically, we are using fine-tuning and freezing classifier layer-based methods. In the present scenario, constructed classifiers are binary classifiers; as it is validated in the paper [32], an increase in the number of classes may decrease the throughput, so it is beneficial to construct binary classifiers for low response time.

7.5 Designing and Implementation

7.5.1 Transfer Learning based Domain Adaptive Multimedia Event Processing Engine

A functional model has been designed for the adaptive multimedia event processing engine (shown in Fig. 7.4), consisting of a keyword-based event matcher, decision model, classifier construction model for training with data construction, processing model for testing multimedia events with modern object detection models and available training datasets as resources. The purpose of the various models with their respective details of implementation is briefly discussed below:

Event Matcher analyzes user subscriptions (such as *bus*, *car*, *dog*) and image events and is responsible for detecting conditions in image events specified by user query and preparation of notifications that need to be forwarded to users.

Training and Testing Decision Model is designed to analyze available classifiers and take the *testing* and *training* decisions accordingly. It evaluates the relationship of existing classifiers with new/unknown subscription and chooses the *transfer learning* technique to train a classifier for the intended new subscription.

Classifier Construction Model phase performs the training of classifiers for subscribed classes and updates the classifier in the shared resources after allowed response-time. The two transfer learning options associated with Hypothesis-III used for classifier construction include fine-tuning and freezing layers (detail in Section–7.5.2). The classifier

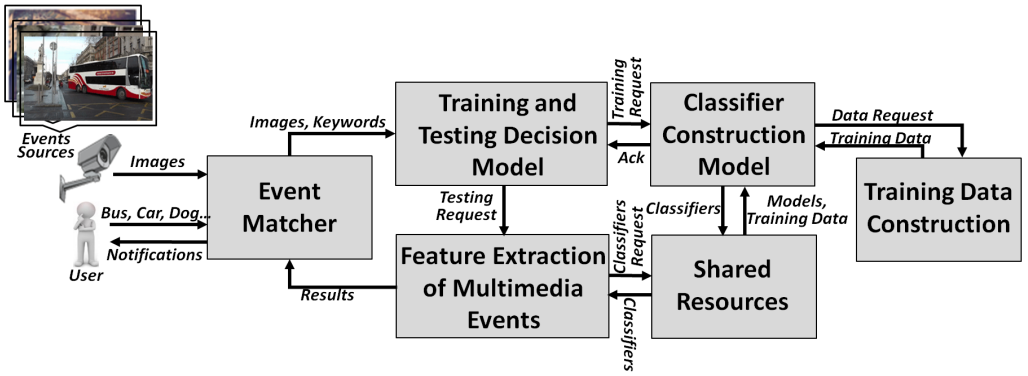


FIGURE 7.4: Transfer Learning based Domain Adaptive Classifier Construction for Multimedia Event Processing

construction module may include the training data construction phase as current object detection datasets [26–28] consist of a limited number of classes. We believe the correct time of adding a classifier to the system is when any subscriber subscribes to it, not when any dataset adds a new class/category to their dataset. Thus, it is more appropriate from the perspective of publish/subscribe to monitor subscribers (not datasets) for adding classifiers.

In *Training Data Construction model*, if a subscriber subscribes for a class which is not present in any smaller object detection datasets (Pascal VOC [26], and Microsoft COCO [27]), then a classifier can be constructed by fetching data from datasets (ImageNet [67], and OID [28]) of more classes using online tools like ImageNet-Utils[1] and OIDv4_ToolKit[2]. Another common approach of online training data construction is to use engines like "Google Images" or "Bing Image Search API" to search for class names and download images. However, accuracy for object detection could be low with this approach; we analyzed this training problem on only images without bounding boxes in Chapter–8.

Feature Extraction of Multimedia Events is responsible for detecting objects in image events using current deep neural network-based object detection models and incorporating new classifiers. Here we utilize image classification models [18, 51, 284] in the backbone network of object-detection models. Image classification models (like DenseNet [149], VGG16 [36], ResNet50 [36], MobileNet [151], etc.) classify a full image with "single label" like "Bus Image", "Car Image", "Person Image" etc., thus applicable for only iconic images. However, images in the real-world are not iconic; they may consist of many objects like road, bus, car, sky, etc., and cannot be labeled in one-class. Therefore utilizing only these light classification models is not enough, and object-detection models employ them in the backbone network for real-time image-event-based applications.

[1] https://github.com/tzutalin/ImageNet_Utils
[2] https://github.com/EscVM/OIDv4_ToolKit

The *Shared Resources* component consists of existing image processing modules and training datasets. We use *You Only Look Once (YOLO)*, *Single shot multibox detector (SSD)*, *and Focal loss based Dense object detection (RetinaNet)* as object detection models [35–37, 149]. We have some base classifiers trained off-line using the established dataset *Pascal VOC* [26], which are assisting us in constructing more classifiers using domain adaptation.

7.5.2 An Approach for Domain Adaptation

The classifier construction decision of the model is based on the suitability of available classifiers for the arrived subscriptions and domain adaptation techniques. The two options of transfer learning used for fine-tuning [285] pre-trained models and freezing backbone layers [286] of similar classifier while training only top dense layers are illustrated in Fig. 7.5. In the first approach, we perform fine-tuning on object detection model pre-trained on ImageNet [67], which uses the technique of back-propagation with labels for the target domain until validation loss starts to increase. Transfer of pre-trained weights over the network and then training classifier for the new subscription assist the proposed model to converge quickly with an increase in accuracy. In the second approach, we use this previously trained classifier to instantiate the network of another classifier required for a similar subscription concept. In this particular scenario [34, 38], we freeze the backbone (convolutional and pooling layers) of the neural network and train only top dense fully connected layers with softmax as output layer, where the frozen backbone is not updated during back-propagation and only fine-tuned layers are updated and retrained during the training of the classifier, this results in less training time with reasonable accuracy.

The current implementation of the proposed model is shown using a general neural networks architecture, which is common among the object detection models (YOLOv3, SSD-300, and RetinaNet) we are utilizing for the purpose of training while using their recommended backbones *DenseNet*, *VGG16*, and *ResNet50* [36, 37, 149] (which could be changed in future). The decision for the construction of a classifier for "bus" either from pre-trained models (by fine-tuning) or from a "car" classifier (by freezing) could be taken with the help of computation of a threshold along the dimensions of accuracy, response time, and similarity. Presently we are using the *path* operator as a WordNet relatedness measure [29] for the computation of similarity among subscriptions, which could be replaced in the future with more accurate measures using image-feature-based domain-specific ontologies depending on the utility of applications.

Algorithm 5 : Domain Adaptive Classifiers based Multimedia Event Processing Engine

Input: $Subscriptions\ (S)\ :\ \{S_1,\ S_2, ...\ S_s\}$, $Classifiers\ C\ :\ \{C_1, C_2, ...C_c\}$, $Pretrained\ Model : P_{Image\_Net}$, and $Image\ Stream : SI$

Output: Subscriber Notifications

while true **do**
  $IE \leftarrow Fetching\_Image(SI)$
  for $k = 1$ to s **do**
    if $(C_k\ \notin\ C)$ **then**
      if $(C\ \neq\ \phi$ **or** $P\ \neq\ \phi)$ **then**
        $C_k \leftarrow domain\_adaptation(S_k, C, P_{Image\_Net})$
      else
        $C_k \leftarrow train\_scratch(k)$
      end if
      $C \leftarrow C_k$
    end if
    $objects \leftarrow object\_detection(IE,\ C_k)$
    if $(S_k\ \in\ objects)$ **then**
      $notify(s_j)$
    end if
  end for
end while

Algorithm 6 : Domain Adaptation

Input: $Subscription\ :\ S_k$, $Classifiers\ C\ :\ \{C_1, C_2, ...C_c\}$, $Pretrained\ Model\ :\ P_{Image\_Net}$

Output: $Classifier : C_k$

$sim \leftarrow 0$ {similarity}
$th \leftarrow 0.01$ {threshold}
$C_k \leftarrow \phi$
$C_{sim} \leftarrow \phi$
for $i = 1$ to c **do**
  $cur\_sim \leftarrow word\_net\_sim(path, classname(C_i), S_k)$
  if $(cur\_sim \geq th$ **and** $cur\_sim > sim)$ **then**
    $sim \leftarrow cur\_sim$
    $C_{sim} \leftarrow C_i$
  end if
end for
if $(C_{sim}\ =\ \phi)$ **then**
  $C_k \leftarrow train\_by\_finetuning(P_{Image\_Net})$
else
  $C_k \leftarrow train\_by\_freezing(C_{sim})$
end if
return C_k

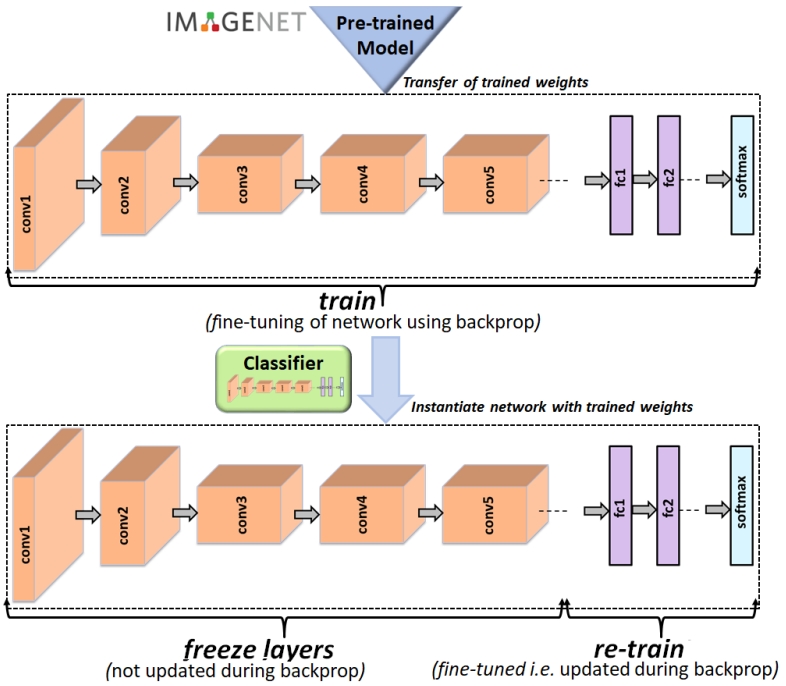


FIGURE 7.5: Techniques used for Transfer Learning

7.5.3 Domain Adaptation based Multimedia Event Processing Algorithms

The implementation of the proposed framework with domain adaptation is detailed in Algorithm 5 and 6. For analyzing image events, firstly, we identify that a classifier for a particular subscription/concept is available or not. In case of unavailability, the model chooses the option of domain adaptation if either the pre-trained model or classifier set is non-empty. Otherwise, in the worst case, training from scratch for the particular classifier is also an option. When the classifier becomes available, we detect objects from image events, and results get sent for the notifications.

The procedure for domain adaptation computes the similarity of subscription with existing classifiers using their class names and WordNet relatedness measure *path* [29], and identify the most similar classifier higher than the given value of threshold (0.01) for training. Lastly, the module chooses the training technique and returns the newly constructed classifier.

7.6 Evaluation

7.6.1 Evaluation Methodology

To test Hypothesis-III, we first compare object detection models on applying domain adaptation techniques using mean average precision (mAP) and response time as performance metrics. Then we present an empirical analysis of the transfer of knowledge on current object detection models and choose the most suitable model. We also show the trade-off of performance with response-time on domain transfers using the selected model and transfer learning technique. Lastly, we report the average accuracy and response time of our proposed framework for seen, unseen, and combination of them; and compare them with existing domain-specific models that recognize only specific seen (known) objects. Since accuracy is not the only measure for analyzing machine learning-based models, we also report the updated confusion matrix (like Chapter–6) relying on the number of missed concepts using snapshots every 15 min in Section–7.6.3.3.

In the experiment setup, I first use Pascal VOC [26] classes for the construction of base classifiers offline to simulate seen subscriptions. Although Pascal VOC is among the established accurate datasets for object detection, it consists of only 20 classes. Thus, I include OID [154] classes using its online data collection toolkit, which assists in finding more combinations of related categories/classes for unseen subscriptions. Specifically, I chose cat, dog, cricket ball, laptop, car, bus, mango, and football classes, because only these were the classes for which I was getting the most distinguished semantic similarity scores. For instance, similarity scores for mango–laptop is 0.08, dog–cat is 0.2, cricket_ball–football is 0.33 and bus–car is 0.5 using *path* operator of WordNet. We used the all training and validation images available for these classes from Pascal VOC and OID datasets for the training of binary classifiers. The total number of training images with bounding box annotations used for cat, dog, cricket ball, laptop, car, bus, mango, and football classes are 1804, 2204, 95, 5528, 2820, 847, 126, and 4339. The available testing images that we used to measure the performance for the same classes are 384, 538, 15, 355, 1588, 256, 23, and 413 respectively.

7.6.2 Evaluation Metrics

We use the following evaluation metrics drawn from the literature [287, 288] for domain adaptation:

- *Response Time* is the time difference between the time subscription arrived and the time at which the system is ready to notify the subscriber.

- *Confusion Matrix* contains information about actual and predicted classifications done by a classification system [271].
- *Accuracy* represents the ratio of correct number of predicted observations (True Positives and True Negatives) to the total number of observations (True Positives, False Positives, True Negatives, and False Negatives).
- *Transfer Loss* is the difference between the error on target data of a model trained on source data (*transfer error*) and the error on target data of a model trained on target data (*baseline in-domain error*), i.e.

$$t(S,T) = e(S,T) - e_b(T,T) \tag{7.1}$$

- *A-Distance* is an approximate distance (known with Distribution Discrepancy) is defined as

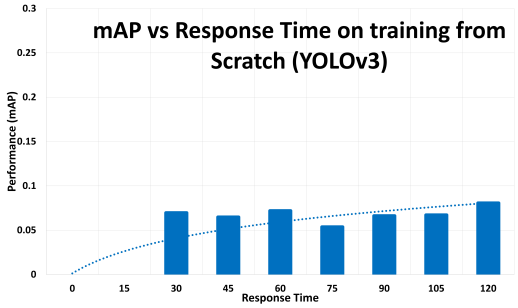
$$d_A = 2(1 - 2\epsilon) \tag{7.2}$$

  where ϵ is the generalization error of a classifier trained on binary problem of discriminating source and target domains.

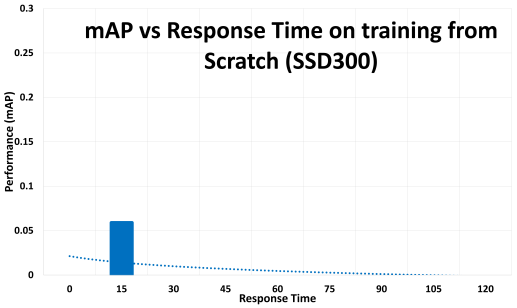
7.6.3 Experiments and Results

7.6.3.1 Performance–Response-Time Trade-off of Object Detection Models

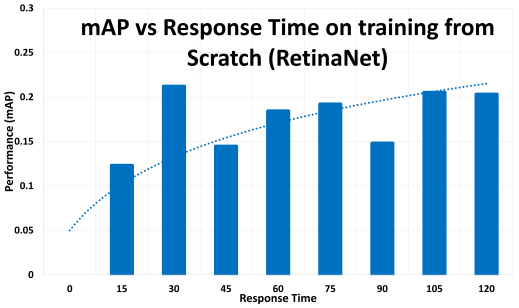
We evaluate transfer learning techniques on different object detection models (YOLOv3 [35], SSD-300 [36], and RetinaNet [37]) to analyze which classifiers can perform well on applying what type of training (scratch, fine-tuning, and freezing layers) and prove Hypothesis-III. The results of performance with *response time* trade-off are shown in Fig. 7.6. We use mean average precision (mAP) for performance, which is the standard method of evaluating neural network models [22]. Firstly, it represents the trade-off on the arrival of a completely new subscription, when there is no possibility of domain adaptation (please refer to Case 2b in Section–2.6), for the training time of 120 min.



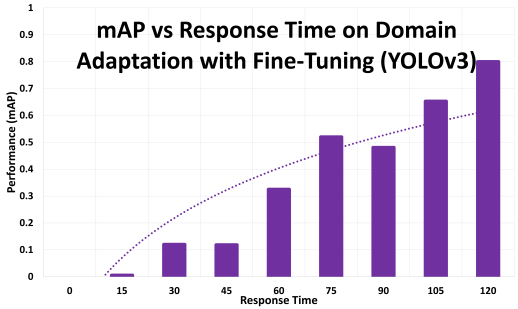
(a) Training from Scratch (YOLOv3)



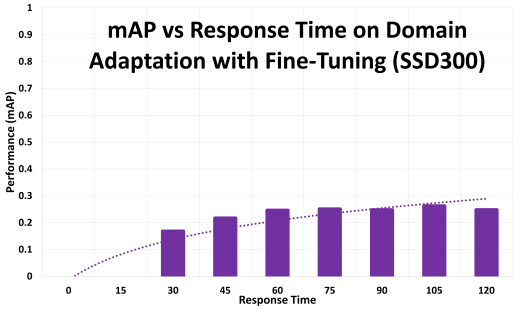
(b) Training from Scratch (SSD300)



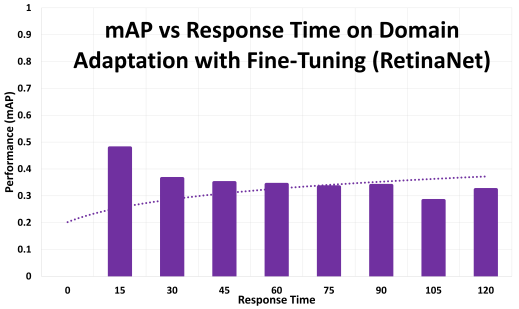
(c) Training from Scratch (RetinaNet)



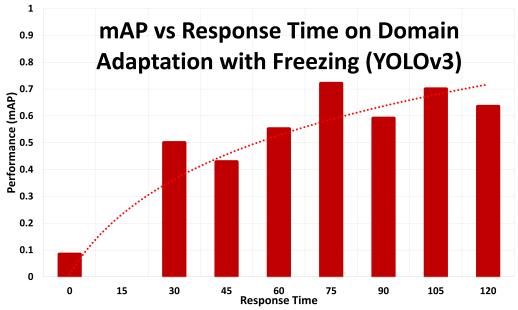
(d) Training by Fine-Tuning (YOLOv3)



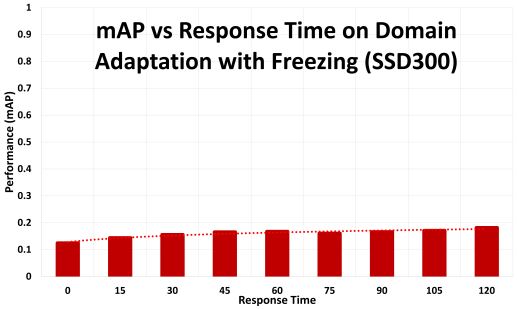
(e) Training by Fine-Tuning (SSD300)



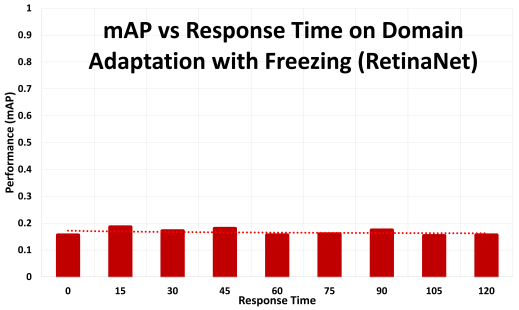
(f) Training by Fine-Tuning (RetinaNet)



(g) Training by Freezing (YOLOv3)



(h) Training by Freezing (SSD300)



(i) Training by Freezing (RetinaNet)

FIGURE 7.6: Performance vs Response Time with and without Adaptation

TABLE 7.2: Evaluations on Domain Adaptation on multiple training techniques

Object Detection Model	**Training from Scratch** $t_{scratch}$			**Training by Fine-Tuning** $t_{fine-tune}$			**Training by Freezing** t_{freeze}			**Frame Rate (in fps)**
	α	β	θ	α	β	θ	α	β	θ	
YOLOv3	0.01	0.09	0.0007	0	0.75	0.0062	0.15	0.79	0.0053	114 fps
SSD300	0	0	0	0.05	0.31	0.0022	0.14	0.19	0.0004	21 fps
RetinaNet	0.09	0.22	0.0011	0.27	0.35	0.0007	0.17	0.16	-0.0001	7 fps

In this case, all models are trained from scratch without the use of any pre-trained model. We can observe the performance of RetinaNet (Fig. 7.6(c)) is higher than other object detection models and the SSD model (Fig. 7.6(b)) is very difficult to converge with training from scratch, thus resulting in worse performance. In contrast, the performance of YOLOv3 (Fig. 7.6(a)) is also low. However, by choosing $t_s = 30\ min$ using RetinaNet, we can reach accuracy $\sim 77.10\%$ with precision ~ 0.21.

The performance of RetinaNet and SSD are better than YOLOv3 in the initial ($<$ 30 min) time of training for both cases of fine-tuning (Fig. 7.6(d), 7.6(e), and 7.6(f)) and freezing (Fig. 7.6(g), 7.6(h), and 7.6(i)) layers (Case 2a in Section–2.6). However, there is a sudden rise in performance of YOLOv3 in the first few minutes, signifying its higher slope in terms of short time training compared to other object detection models. We can easily observe that all object detection models with the fine-tuning technique perform better than the adaptation technique of freezing layers for a long training time (i.e., $>$ 120 min). However, for short training time (i.e., $\sim$ 30 min), YOLOv3 with freezing technique is performing the best.

We can also observe all object detection models (YOLOv3, SSD, and RetinaNet) on direct domain transfer provide mAP of 0.1, 0.12, and 0.16, respectively, within a response time of 0 min. In contrast, other techniques (finetuning of pre-trained model and training from scratch) possess an mAP of 0, which validates our hypothesis.

The results of various parameters (detailed in Fig. 2.3(b)) derived using trend lines of Fig. 7.6, for different types of training ($t_{scratch}$, $t_{fine-tune}$, & t_{freeze}) are shown in Table 7.2, which also supports the fact of achieving high initial performance, high rate of change, and high final precision achieved in given training time, on domain adaptation of classifier. The recorded frame rates on our resources for YOLOv3, SSD300, and RetinaNet are 114 fps, 21 fps, and 7 fps, respectively, where fps represent the number of frames/images processed per second. We can conclude that adaptation via *freezing* layers can provide admissible performance (i.e., accuracy $\simeq 95.14\%$ with precision $\simeq 0.50$ using YOLOv3) in initial training time ($t_{da} = 30$ min) as compared to *fine-tuning* of a

TABLE 7.3: Detection mAP on Specific Domain Transfers using different Domain Adaptation techniques

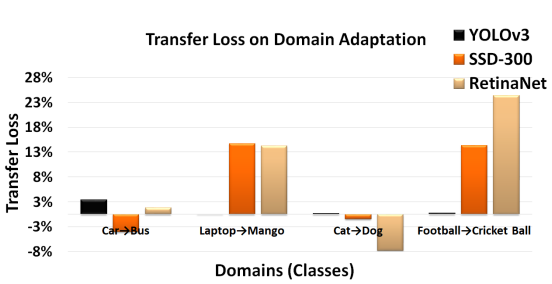
Classes	**Semantic Similarity Score**	**Method of Domain Adaptation**	**YOLO**	**SSD300**	**RetinaNet**
Mango ← Laptop	0.08	Laptop Detector tested for Mango (**baseline**)	NaN	0.0047	0.0046
		Mango Detector from pre-trained model	NaN	0.1439	0.1667
		Mango Detector from Laptop Detector	0.2000	0.0818	0.0973
Dog ← Cat	0.20	Cat Detector tested for Dog (**baseline**)	0.0000	0.2123	0.2446
		Dog Detector from pre-trained model	0.5254	0.2120	0.2159
		Dog Detector from Cat Detector	0.6875	0.2504	0.2307
Cricket_Ball ← Football	0.33	Football Detector tested for Cricket ball (**baseline**)	0.0000	0.0000	0.0111
		Cricket ball Detector from pre-trained model	NaN	0.00	0.0375
		Cricket ball Detector from Football Detector	0.0000	0.0000	0.0120
Bus ← Car	0.50	Car Detector tested for Bus (**baseline**)	0.1683	0.0371	0.0668
		Bus Detector from pre-trained model	0.7213	0.0938	0.1110
		Bus Detector from Car Detector	0.5821	0.1127	0.0808

NaN: Not a Number (could be interpreted as mAP=0 because of no detections)

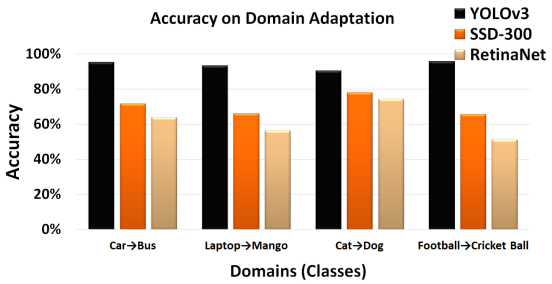
pre-trained model, which is crucial to know before choosing either pre-trained model or nearest classifier.

7.6.3.2 Empirical Analysis for Domain Shift

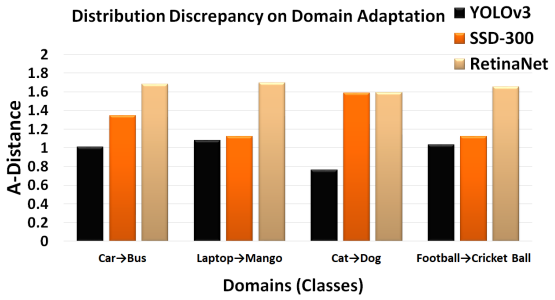
Table 7.3 shows four examples of classes on domain transfers with different similarity scores. We determine similar classifiers using the *path* operator of WordNet [29]. Here, we show simple baseline performance where the nearest neighbors can detect an unseen class without training, thus resulting in low mAP but zero response-time. Besides testing on baselines, we show performance on domain transfers from pre-trained models and similar class detectors. We can conclude that adapting from one domain (class) to



(a) Transfer Loss



(b) Accuracy



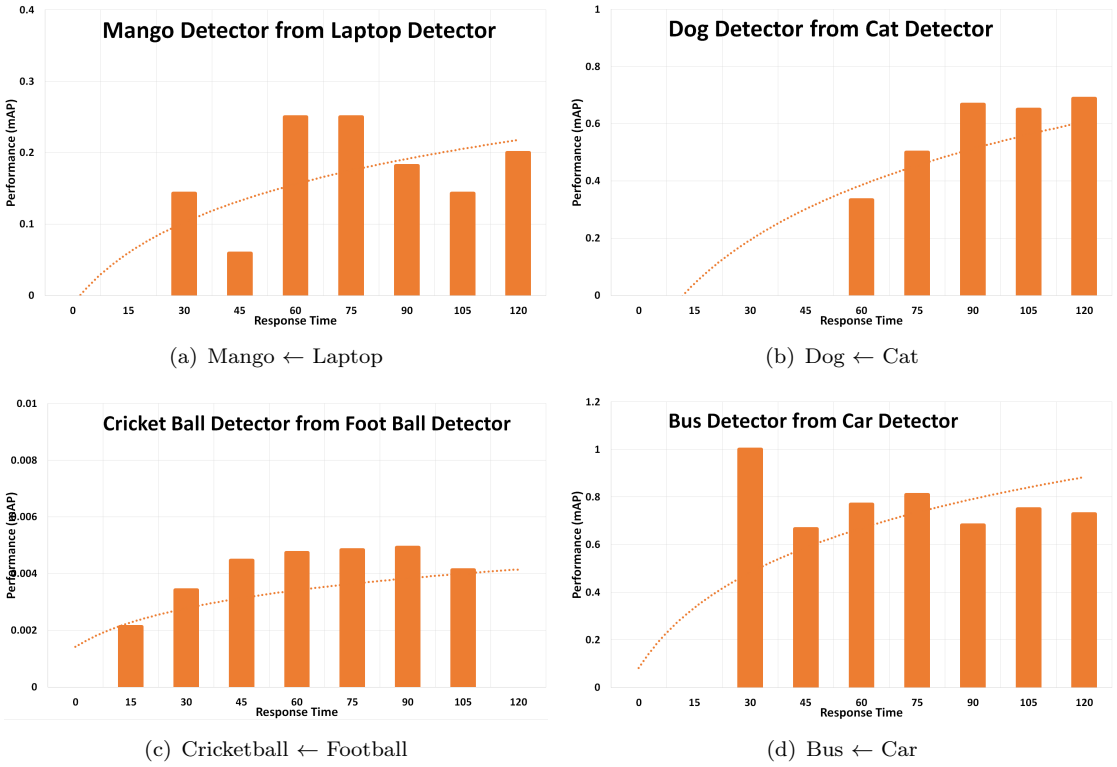
(c) A-Distance

FIGURE 7.7: Analysis for Domain Shift

another mostly yields high-performance results in low response-time as compared to fine-tuning of detectors on the pre-trained model (like ImageNet).

We also present an empirical analysis to know how well the transfer works on current object detection models and choose the most suitable model. Fig. 7.7 analyzes *Transfer Loss*, *Accuracy*, and *Distribution Discrepancy*, during the domain shift of subscriptions on object detection models. The standard domain adaptation metric "transfer loss" has been evaluated on four domain transfers (varies from closely related domains to not associated domains), depicted in Fig. 7.7(a). The transfer achieved by YOLOv3 is better than other object detection models in the case of *football* to *cricket ball* and *laptop* to *mango* domain transfers. Here, the transfer loss only indicates how well the transfer works on multiple domains, and its lower values are more recommended [287, 288]. However, the best transfer (i.e., least transfer loss) is achieved by the RetinaNet model on *cat* to *dog* class transfer. Similarly, the Single Shot Detection (SSD) model achieves its best in transferring *car* to *bus* domain transfer. Interestingly, the values of transfer loss using models SSD and RetinaNet on other domain transfers are quite high, thus directing us to evaluate the *accuracy* of same domain adaptations.

The transfer accuracy achieved by object detection models on the same classes of domain transfers (discussed for the evaluation of transfer loss) is shown in Fig. 7.7(b). We can clearly see that all object detection models (YOLOv3, SSD-300, and RetinaNet) are



(a) Mango ← Laptop

(b) Dog ← Cat

(c) Cricketball ← Football

(d) Bus ← Car

FIGURE 7.8: Performance vs Response Time with Domain Adaptation

able to provide high accuracy on applying transfer learning techniques; however, the YOLOv3 achieves the best accuracy on all domain transfers.

In order to realize the variation of approximate distance (i.e., Distribution Discrepancy) among different domains, we have trained few binary classifiers that can classify source-target pair of classes like *cat and dog*, *car and bus*, *football and cricket ball*, and *mango and laptop*. We can see in the results (Fig. 7.7(c)) that distribution discrepancy (lower is better) for YOLOv3 is relatively smaller among most of the domain transfers than other object detection models, which suggests that the YOLOv3 neural network closes the cross-domain gap more effectively, which also explains its better accuracy than other object detection models on domain adaptation.

7.6.3.3 Simulation on Proposed Model

Results on Domain Adaptation As previous results of high performance and domain shifts are in favor of YOLOv3 with freezing layer-based transfer learning technique, we have shown trade-off of performance with response time for domain transfers (Mango ← Laptop, Dog ← Cat, Cricket Ball ← Football, and Bus ← Car) in Fig. 7.8. We can observe that even with a training time of only 120 min, we get performance up to 0.20, 0.69, 0.004, and 0.73 on each domain transfer. Interestingly, for short response time (15

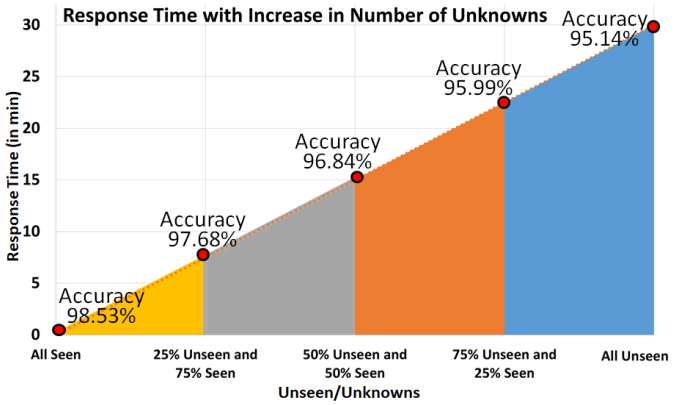


FIGURE 7.9: Response Time with Unseen Subscriptions

or 30 min), we also get reasonable performance in all transfers except Dog ← Cat. Thus we can choose response-time (∼ 30 min) to provide reasonable accuracy for such unseen subscriptions.

Confusion Matrix Table 7.5 summarizes the snapshots of the confusion matrix at the interval of every 15 min, as accuracy is not the complete measure for analyzing machine learning models. Here we expect the values of true positives and true negatives (shown in dark gray color) to increase with time while false positives and false negatives (shown in light gray color) decrease with time. This is true for most cases with a few misleading values (due to less training and testing data available for classes like *mango* and *cricket ball*). Other than the problem of inadequacy of data, we are training all classifiers less than 10 epochs in-order to reduce the response time. However, the recommended training time of existing neural network-based models is not less than 100 epochs (i.e., $>$ 1 day).

Seen/Unseen Domains An average response time of the proposed model for seen subscriptions depends only on the testing time (∼0.01 min) of the object detection model. Response time for unseen subscription also includes training via domain adaptation. Due to this reason, accuracy (98.53%) for seen subscriptions is even higher than accuracy (95.14%) for unseen subscriptions. As a result, response time will increase, and accuracy will decrease with the number of unseen subscriptions. An analysis by a varying number of seen/unseen subscriptions is shown in Fig. 7.9. For example, when the number of unseen subscriptions increases from 0 to 25% of the total number of seen and unseen subscriptions, then response-time will always be between 0 to 7 min and accuracy between 98.53% to 97.68%. Similarly, for 50% of unseen subscriptions w.r.t,

TABLE 7.4: Comparison of Proposed with Existing Model(s)

Approach	**Example of Subscriptions**	**Performance**	
		Response Time	**Accuracy**
Existing Domain Specific Models	Vehicle Detection [233] (Seen)	0.001 min	97.30%
	Firearm Detection [234] (Seen)	0.0001 min	94.00%
	Stolen Objects [235] (Seen)	0.0007 min	93.58%
	Car Parking Vacancy [236] (Seen)	0.17 min	97.90%
	Traffic Light, Key, Pedestrian, Ball, Bag etc. (Unseen)	∞	0.00%
Proposed Model	Car, Football, Cat, Laptop, Cake, Flower etc. (Seen)	0.01 min	98.53%
	Bus, Dog, Person, Cricket ball, Coin, Suitcase etc. (Unseen)	29.99 min	95.14%

the total number of subscriptions will increase response-time up to 15 min and accuracy to 96.84%.

Table 7.4 provides a comparison of average accuracy and response time of proposed with existing models by considering their best performance. We can observe that existing domain-specific models are designed only to detect specific objects and answer such seen (known) subscriptions in low response time; however, they fail to process any unseen (unknown) subscription of a different domain. The proposed model can achieve an accuracy of 95.14% even when all concepts are unseen by taking an average response time of ∼30min.

Other than the factors discussed in experiments, few external factors can also impact the performance of Model-III. For instance, execution environment, available resources, quality of image, number of images available for domain adaptation, the similarity of unseen class with seen class, etc. The number of resources and execution environment can increase/decrease the response time further. On the other hand, the quality of training data can improve the accuracy of the model. Moreover, subscriptions of users cannot be altered at the administrative level; subscribed unseen classes can be similar or utterly dissimilar to seen classes. Thus, Model-III performance may change with change in the application domain.

7.7 Conclusions and Discussion

In this chapter, I analyzed the problem of processing multimedia events that include a large number of seen/unseen concepts belonging to the same or multiple domains, using the online construction of classifiers while minimizing response time. Discussion

on related work reveals that domain transfer approaches are useful for high accuracy and could help support large-scale vocabulary but leaves the gap for analyzing training time. I proposed a multimedia event processing framework with the feature of inter/intra domain adaptation among subscriptions by utilizing transfer learning-based techniques and object detection models. I analyzed the trade-off between performance and response-time, which also includes training time, thus providing a holistic view of the comparison of DNN based models. Such trade-offs validate the Hypothesis-III by providing some accuracy even at zero response-time. Moreover, experiments have been conducted on the various shift of domains among subscriptions to determine the minimum permissible response-time and best detector on which transfer works well. The proposed approach can achieve accuracy ranges from 95.14% to 98.53% within $\sim$ 0.01 min to $\sim$ 30 min of response-time using the YOLOv3 object detection model even when all subscriptions are unseen (unknown) for the system.

Another specific problem that originated in this chapter is the requirement of annotated bounding boxes for online domain adaptation of models. In the next Chapter–8, we extend our model for semi-supervised learning to reduce the need for labeled data to process unseen concepts.

TABLE 7.5: Confusion Matrix with Response Time for the Domain Adaptive Multimedia Event Detection Model using YOLOv3

Time (in min)	Confusion Matrix		Subscriptions (Expected)															
			Cat		Dog		Cricketball		Laptop		Car		Bus		Mango		Football	
			P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N
15	Predictions	P'	370	12262	0	0	2	938	199	5942	1541	15509	0	0	26	2823	220	7503
		N'	0	4620	530	4952	5	399	0	224	0	4177	254	4952	0	390	0	195
30		P'	0	9	0	0	2	582	14	3	258	713	1	0	15	803	34	10
		N'	370	4952	530	4952	5	399	185	387	1283	4760	253	4951	11	394	186	367
45		P'	1	4	0	0	2	446	70	175	779	3516	6	1	18	676	179	1117
		N'	369	4951	530	4952	5	399	129	339	762	4571	248	4946	8	392	41	233
60		P'	2	12	1	2	1	270	121	737	1240	8508	10	6	13	290	155	611
		N'	368	4951	529	4951	6	400	78	292	301	4347	234	4935	13	394	65	257
75		P'	0	0	5	8	1	270	96	162	2	0	34	8	13	262	209	3588
		N'	370	4952	525	4947	6	400	103	314	1539	4950	220	4924	13	395	11	205
90		P'	0	0	8	16	3	299	109	545	43	2	28	5	13	253	206	1370
		N'	370	4952	522	4944	4	398	90	304	1498	4916	226	4925	13	394	14	208
105		P'	3	0	15	9	1	241	88	321	94	4	36	12	11	205	196	567
		N'	367	4949	515	4938	6	400	111	326	1447	4869	218	4918	15	395	24	217
120		P'	9	2	11	5	0	88	124	318	85	12	39	28	9	168	188	470
		N'	361	4943	519	4942	7	401	75	288	1456	4878	215	4919	17	395	32	224

Chapter 8

Domain Adaptation based Multimedia Event Detection without Bounding Boxes

8.1 Introduction

In previous Chapter-7 I proposed a domain adaptive classifier construction approach with knowledge transfer from seen to unseen concepts, for minimizing the response time while achieving high accuracy. However the major limitation of that approach is that it requires annotated bounding boxes for the online training of unseen classes. Training of object detection models using only image-level labels is an emerging challenge in computer vision, as obtaining object bounding box annotations is an extremely time-consuming task. In the current scenario, we have object detection datasets available consisting of a small number of classes or a small number of images per class. Moreover, image-level labels are comparatively easy to acquire, many classes can be covered easily using image classification datasets (like ImageNet [67]) or the web. Recently, classifiers to detector conversion methods [3, 30, 66] have shown promising results for the training of unseen concepts without bounding boxes. However, these methods have not considered the long training time and only handle finite number of classes.

In this chapter I tests the research Hypothesis IV "*If an adaptation of classifier into detector eliminates the need of bounding boxes as well as transferring of knowledge from one domain to another speed-up the training; and a detector gets constructed from classifier with the help of transfer of knowledge from visually/semantically similar classifier; then that detector will take less time to train for unseen classes and eliminate the requirement of bounding boxes*". I formulate this problem in the final specific Research Question

3(b) as "*How can we answer multimedia event based queries online consisting of unseen subscriptions (unbounded vocabulary), using task as well as visual domain adaptive classifier construction approach with knowledge transfer from seen subscriptions (bounded vocabulary) while eliminating the requirement of bounding box annotations availability, achieving high accuracy, and minimizing the response time?*" discussed in Section–8.2. A brief background on weakly supervised learning and classifier to detector conversion methods is given in Section–8.3, which shows gaps of lack of support for large vocabulary and least focus on training time.

In this work, I propose an "Unseen Detector" that can be trained within a very short time for any possible unseen class without bounding boxes with competitive accuracy. The proposed framework is shown in Figure 8.4 with detail in Section–8.4. I build approach on "Strong" and "Weak" baseline detectors, which I trained on existing object detection and image classification datasets, respectively. Unseen concepts are fine-tuned on the strong baseline detector using only image-level labels and further adapted by transferring the classifier-detector knowledge between baselines. I use semantic as well as visual similarities to identify the source class (i.e. Sheep) for the fine-tuning and adaptation of unseen class (i.e. Goat).

My model is trained on the ImageNet classification dataset for unseen classes and tested on an object detection dataset (OpenImages) consisting of the same classes. The model achieves a mean average precision (mAP) of 19.82 within 5 minutes of training, where existing frameworks could take >5.5 hours to attain a similar mAP (discussed in Section–8.5). Quantitative and qualitative results demonstrate that proposed model is suitable not only for the iconic images of ILSVRC but also for object detection datasets and images from the web for any unseen concept. Finally we concludes and discusses the drawbacks of our "UnseenNet" in Section–8.6.

8.2 Problem Overview

In the case of our fast training detector for unseen concepts without bounding boxes, we assume that we have access to the object detection datasets (i.e., training images with bounding box annotations for the small number of classes) and image classification datasets (i.e., training images with only image-level labels for the large but finite number of classes). Our objective is to train detectors for any possible unseen concept (i.e., an infinite number of classes) without bounding box annotations within a limited amount of time. This is quite different from the existing classifier to detector knowledge transfer based methods [3, 30], where a limited number of *unseen* classes are trained for a more extended period of time with a focus on improving only on accuracy. In such cases

where we already know *unseen* classes and are allowed to train for longer training times, a better solution is to train unseen classes with high accuracy by taking more time while making bounding box annotations available. Below we discuss few preliminaries, motivational scenarios, and this problem in detail.

8.2.1 Preliminaries

LSDA: Large Scale Detection through Adaptation (LSDA) [3, 66] converts image classifiers into object detectors by transferring knowledge between pair of classes for which we have both classifiers and detectors, where paired relationships identify using semantic and visual similarities between classes.

Domain Adaptation with Task Transfer:

- Adaptation of classifiers into detectors: We consider classification (full image recognition) as Task-1 (our source domain) and detection (localized recognition) as Task-2 (our target domain) and cast the transformation between the source and target domain as a domain adaptation problem (like baseline LSDA).
- Adaptation between Visual Domains: Allows Inter/Intra domain adaptation. For example, Bus→ Car or Person→Pedestrian, where traffic management (car, bus, pedestrian, bike) and parking management (car, taxi, bike, person) are two different domains.

Similarity:
For the adaptation of seen class into unseen class, we use visual and semantic similarity measures:

- Visual Similarity: Visual similarity measurements are often computed using the minimal Euclidean distance between feature distributions of the last layers of CNN [30].
- Semantic Similarity: It is a well-established field in the Natural Language Processing community. WordNet [29] is the popular lexical database of semantic relations between words, and we consider *path vector* as the most suitable semantic similarity measure (correlated with visual similarity) in this study.

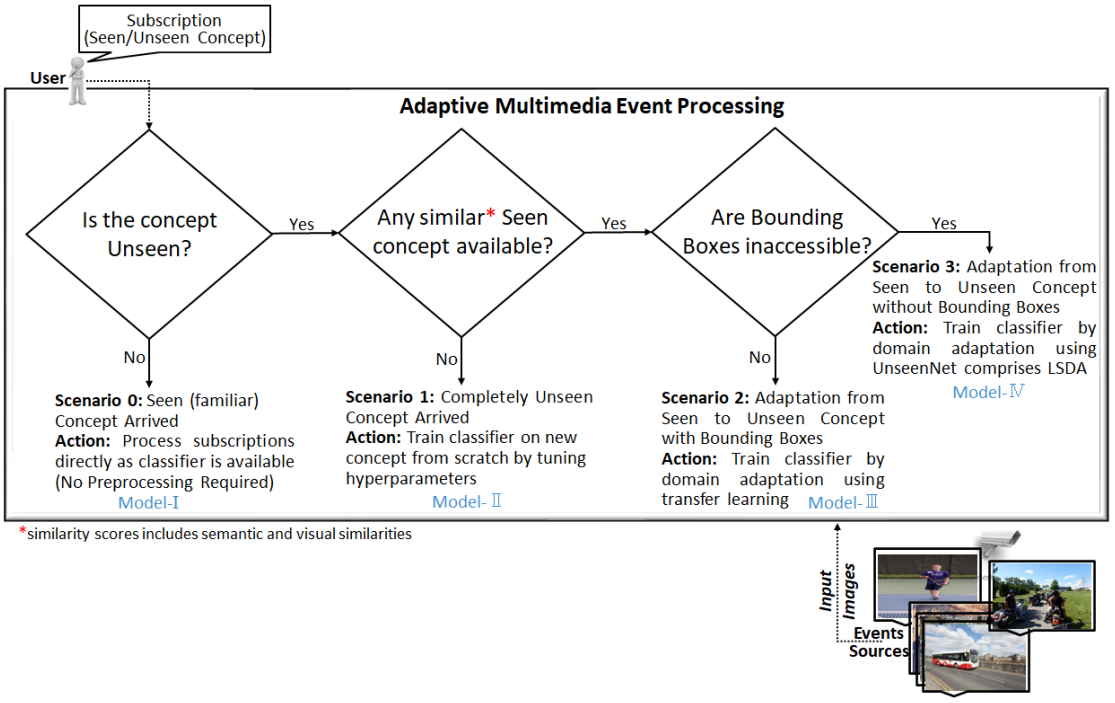


FIGURE 8.1: Scenarios for Multimedia Event Processing adhering to Seen/Unseen Concept Problem

8.2.2 Motivational Scenarios

Consider the baseline scenarios of detecting seen/unseen concepts for analyzing multimedia events, as shown in Fig. 8.1. The last scenario is also associated with the partial unseen concept, but it removes the limitation of the previous Scenario-2, where we need annotated bounding boxes to train models. Presently, most of the object detection datasets have limited vocabulary; thus, we cannot provide bounding boxes for a large number of unseen concepts. The problem analyzed in this work represents the case of classifier not available for subscription (unseen concept), and object-level annotations are also not available, presented in Section–1.4.

Suppose a user subscribes for an unseen class "goat" and we have a detector available for seen class "sheep" which is visually and/or semantically similar to *goat* class. The previous adaptation model can then adapt sheep detector into goat detector using domain adaptation techniques (used in Chapter–7). However, in the present case, we have only images and no bounding boxes (unlike previous work), so we cannot use conventional transfer learning methods. An example of an image and object-level annotations is shown in Fig. 8.2. If we have an adaptation model that includes a classifier to detector conversion mechanism, then we could answer such "unseen" subscriptions having no annotated bounding boxes by training *classifier* on image-level labels and convert them into *detector* by knowledge transfer using visually/semantically related *seen* classes.

(a) Image-Level Annotations (b) Object-Level Annotations

FIGURE 8.2: Annotations with/without Bounding Boxes

8.2.3 Problem Statement

Can we train detectors for any possible unseen concept (with only image-level labels) within a limited amount of time while utilizing classifier to detector conversion methods [3, 30] and existing limited vocabulary based object detection and image classification datasets?

8.3 Background and Related Work

The fundamental challenge in training object detection models is the need to create a large number of annotated images. Moreover, if we look towards large scale human-level category detection systems, it is impractical to collect a large quantity of bounding box labels for millions of categories. Considerable research [30, 61, 66, 289, 290], including LSDA, cast the task of transformation as a domain adaptation problem by considering images with only labels as the source domain and the images with bounding boxes as the target domain.

8.3.1 Weakly Supervised Object Detection (WSOD) with Knowledge Transfer

Recently, weakly supervised learning [291, 292] is emerging as a possible solution for large-scale unseen concepts. Uijlings et al. [289] proposed a revisit knowledge transfer for detectors training in the weakly supervised settings and outperformed all the baselines. Similarly, a mixed-supervised approach [290] is also presented with the condition of strong categories and weak categories have no overlap. Bilen et al. [293–295] proposed different solutions for the weakly supervised object detections using deep detection architecture, convex clustering, and posterior regularization. Alexander et al. [296] improved localization by introducing distractor labels with objects (e.g. trains on tracks). Similarly, Wang et al. [297] proposed a cluttered backgrounds based approach

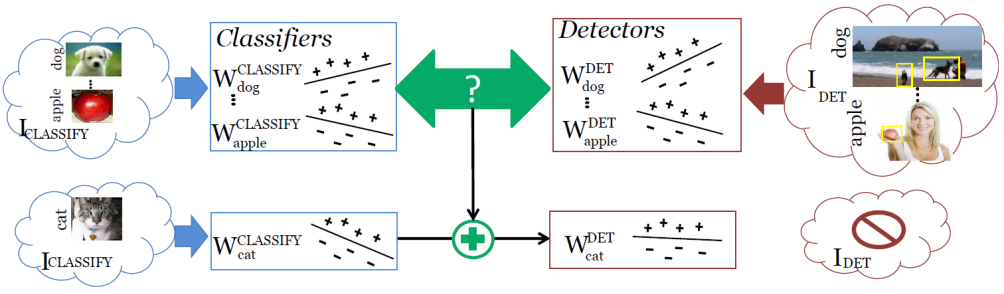


FIGURE 8.3: Conceptual Representation of LSDA: Large Scale Detection through Adaptation [3]

and a transfer learning based model [298] uses appearance similarity with ranking. Such approaches are designed for limited classes and cannot incorporate new classes that have no pre-trained models.

Weakly supervised learning also formulated as a Multiple Instance Learning (MIL) problem. A novel $WSOD^2$[299] model uses bottom-up object evidences and top-down classification output with an adaptive training mechanism. Model uses VGG16 [18] backbone, pre-trained ImageNet [67] for initialization. However our model uses the much faster MobileNetv3 [151] as backbone and our own trained "Strong Baseline Detector" for the initialization. Similarly a MIL based approach appear in literature [300, 301] for the weakly supervised object localization. Most of the existing approaches [300–303] are evaluated on classes of Pascal VOC [26], disregard the training time, and/or use Fast RCNN [147] as base network. However, Fast RCNN has a longer inference time, and Pascal VOC [26] is known to the computer vision for a long time, its classes shouldn't be considered unseen.

8.3.2 Large Scale Detection through Adaptation (LSDA)

Another category of related work includes the Large scale Detection through Adaptation (LSDA) based approaches [3, 30, 61, 66] that incorporate the knowledge transfer from source to target domain. Such methods consider the difficulty of obtaining labeled images for large numbers of categories as a major barrier of visual recognition systems; thus, we assume it is desirable to bring their abilities to the core of multimedia event processing. It is realized in these methods that image-level annotations are comparatively easy to acquire, and search engines could quickly produce a set of images using corresponding image tags. LSDA [3] method learns to transform an image classifier into an object detector. LSDA learns the difference between the two tasks (classification and detection) and transfers this knowledge to classifiers to turn them into detectors without the need for bounding box annotations. The core idea is shown in Fig. 8.3. LSDA train detectors

(weights) from labeled classification data (left) for several classes and consider some classes (top) that also have detection labels (right), and train their detectors. Then it addresses the problem of classes with classification data but no detection data (shown at the bottom) using the paired relationships between classes for which both classifiers and detectors are available, and transfer that knowledge to the classifier of the bottom to convert it into a detector. The transformation of classifiers into detectors of LSDA follows a transfer learning problem due to statistical distribution differences of training and test domains. This large-scale learning of detectors exploit weak (image-level) and strong (bounding box) labels and transfer learned perceptual representations from related tasks.

LSDA also gets extended to the multiple instance learning (MIL) framework [304] that includes bags defined on both types of data and optimizes the perceptual representation using strong detection labels from related categories. LSDA with MIL demonstrates the more accurate adaptation results on new (weak) categories. However, LSDA provides a detector for the limited number of classes without bounding box annotations with the not straightforward provision of adding unseen classes. Also, there is no control over the training time of LSDA based detectors. Such limitations make existing work the least significant. Other than being trained on a finite number of classes, their significance is hard to judge and needs to be seen from the perspective of the dynamic environment of smart cities. Such limitations make existing work least significant for training new classes.

Tang et al. [30, 61] improve LSDA by incorporating informed visual knowledge and semantic similarities during the transfer process. This work is focused on the hypothesis "visually, and semantically similar categories exhibit more transferable properties than dissimilar categories". For instance, a cat detector constructed from dog classifier and dog detector will be much better than a cat detector built from violin classifier and detector differences. Evaluations of the proposed approach on the ILSVRC2013 [152] detection datasets demonstrate the effectiveness of using visual and semantic similarities by improving detection accuracy over the LSDA baseline. However, following other approaches, training time is not considered in the improved LSDA model, making the traditional argument "object bounding-boxes computation is a time-consuming task" weaker, especially when models are taking a considerable amount of training time.

8.3.3 Gap Analysis

As a result of the analysis of related work, Table 8.1 shows a comparison of existing approaches with mapping of requirements (suggested in Section–2.4). While classifying the related work, we summarize the gap analysis with limitations as follows:

TABLE 8.1: Analysis of Related-Work with identified Requirements for Knowledge Transfer without Bounding Boxes

Category	Approach	**Requirements**			
		High Accuracy for Multimedia Events	Low System Response Time	Support for Large Vocabulary	Maintain-ability
Weakly Supervised Object Detection with Knowledge Transfer	Rivist Knowledge Transfer [289]	Average Accuracy	N.A	N.E	N.E
	Mixed Supervised [290]	Average Accuracy	N.A	N.E	Generalizable to New categories
	Deep Detection [293–295]	Average Accuracy	N.A	N.E	N.E
	Distractor Labels [296]	Average Accuracy	N.A	N.E	N.E
	Cluttered Background Approach [297]	Average Accuracy	N.A	Large Scale Method	N.E
	Appearance Similarity Approach [298]	Higher Accuracy	N.A	N.E	Transferable to unrelated object categories
	WSOD [299]	High Accuracy	N.A	N.E	N.E
	Other MIL based Approaches [300–303]	Low to Average Accuracy	N.A	N.E	Adaptable to new Object Categories
Large Scale Detection through Adaptation (LSDA)	LSDA [3]	Low Accuracy	N.A	Large Scale Vocabulary	Manual Adaptation
	LSDA with MIL [66]	Low Accuracy	N.A	Large Scale Vocabulary	Manual Adaptation
	Semi-supervised LSDA [30, 61]	Low Accuracy	N.A	N.E	N.A

N.A: Not Applicable

- *Weakly Supervised Object Detection with Knowledge Transfer*: Weakly supervised learning approaches, along with multiple instance learning, are highly focused on increasing accuracy. Some of them are also transferable to new object categories. However, the support for large-scale vocabulary is not evaluated. Moreover, the testing or training time of such models are not considered, which are crucial to know for real-time multimedia event processing.

- *Large Scale Detection through Adaptation (LSDA)*: These classifiers to detectors knowledge transfer methods can prove an asset in training classifiers for multimedia event processing. Image-level labels are plenty, and object-level annotations are hard to acquire. Due to the less attention over such knowledge transfer methods, these methods struggle to achieve high accuracy. However, the ability to provide large-scale vocabulary and not a straightforward adaptation approach shows the critical gap that needs investigation. Lastly, techniques for reducing the training time for minimizing the overall response time are not known to date.

8.4 Proposed Approach

We first provide an introduction of LSDA (Section–8.4.1) that we took as a baseline for our work on unseen classes. Then we explain our model (shown in Figure 8.4) with details on training detectors offline for seen concepts (Section–8.4.2.1) and online for unseen concepts (Section–8.4.2.2).

8.4.1 Baseline LSDA

LSDA [3, 66] transforms image classifiers into object detectors in three steps: (1) Training LSDA: Category Invariant Adaptation (includes initialization of detection parameters and network surgery), (2) Training LSDA: Category Specific Adaptation, and (3) Detection with LSDA. The objective of LSDA is to detect K categories while having bounding box annotations for m categories. Consider the set of images with strong labels as B=1,...m and weak labels as A=m,...K where $m \ll K$. First, 8-layer AlexNet [305] is pre-trained on the ILSVRC challenge, the final weight layer (1,000 categories) is then replaced with K classifiers, and fine-tune the whole network on classification data $C = C_A \cup C_B$. The next step is the net surgery on this *classification network* by fine-tuning layers $1-7$ on strongly labeled data (i.e., with bounding boxes) of categories B. The fine-tuning on strong labels also learns a generic *background* category because of bounding boxes. It is important to note that fc_8 layer parameters are *category specific*, while layers $1-7$ are referred to as *category invariant*.

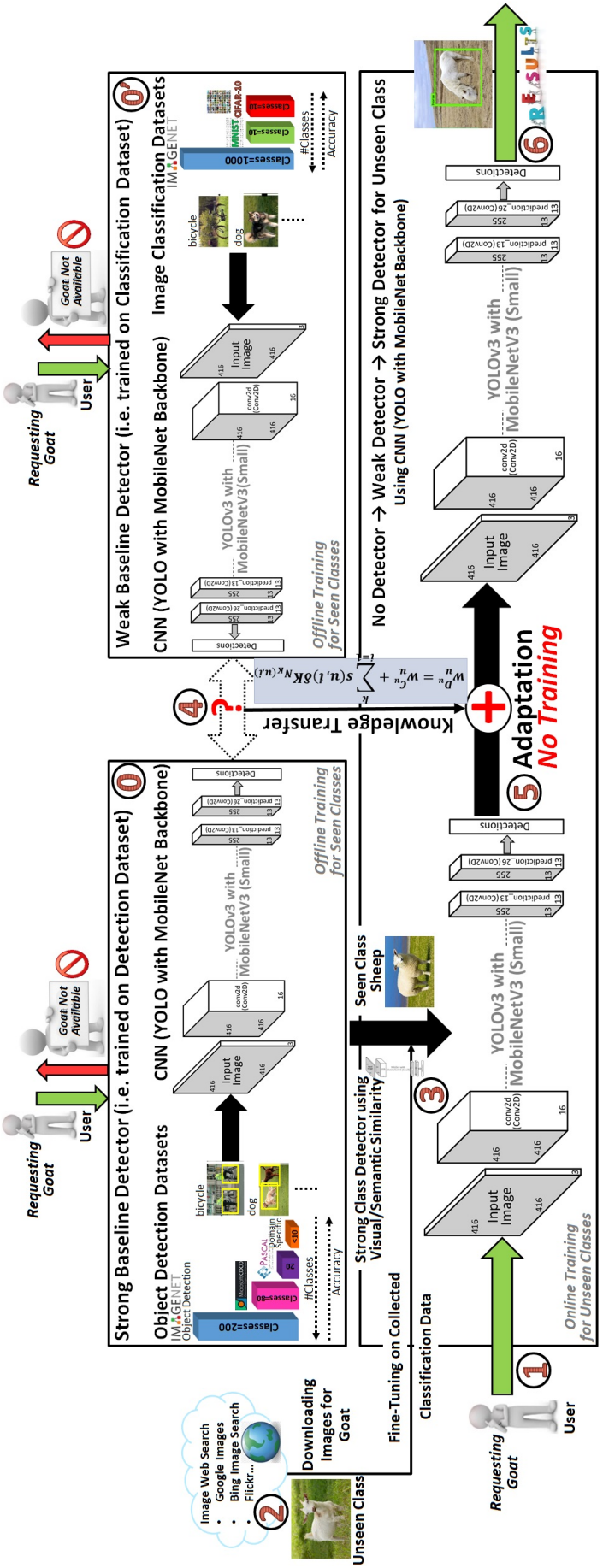


FIGURE 8.4: An illustration of our "UnseenNet" model.

Category-Specific adaptation step is responsible for the final transformation of the classification network into a detection network. For the demonstration LSDA separates the 8th Layer into two components fc_A and fc_B. For categories in set B, the transformation is directly learned by fine-tuning the category-specific parameters fc_B. Suppose the weights of the classification network's output layer are W^c, and the weights of the output layer after adaptation are W^d. LSDA assumes the final detection weights for category $i \epsilon B$ could be computed as ${W_i}^d = W_i^c + \delta_{B_i}$. There is no detection data for categories in set A; thus, LSDA approximated the fine-tuning that could have occurred to fc_A using nearest neighbor categories of set B for each category of set A. The final output detection weights are:

$$\forall_j \in A : W_i^d = W_i^c + \frac{1}{k} \sum_{i=1}^{k} \delta B_{N_B(j,i)} \tag{8.1}$$

where k^{th} nearest neighbor in set B of category $j \in A$ is denoted as $N_B(j, k)$. At the detection time, LSDA directly outputs scores from the softmax "detector', and reduce the training time from 3 days to 5.5 hours.

8.4.2 UnseenNet: Designing and Implementation

We propose an "UnseenNet" detector, which allows a user to construct detectors for unseen classes without the need for detection data (no bounding boxes) within the short training time. Our model is based on making use of existing object detection datasets of bounded vocabulary (consists of *seen* concepts) to construct detectors for *unseen* concepts (i.e., unbounded vocabulary) by using the differences between a weak detector (trained on image classification dataset) and a strong detector (trained on object detection datasets).

An illustration of our "UnseenNet" model is shown in Figure 8.4. We train two separate detectors, "Strong Baseline Detector (0)" and "Weak Baseline Detector (0')" offline using bounding box annotations and image-level labels, respectively. Then UnseenNet follows the below steps:

1. Download images using only image-level labels on request of any *unseen* concept (like a goat).

2. The *strong baseline detector* is then fine-tuned on collected images of unseen concepts by labeling the most semantically similar class (like sheep) with the unseen class name (like a goat). It is worth noting that the minimum value of similarity

among classes can reach up to 0.0357 (for pair of Kite and Bull), and the maximum similarity value could be 1 (for same class names) according to our analysis of WordNet [29] relatedness measure.

3. At this stage, I compute the visual similarity of the constructed unseen class detector (trained on classification data) with seen classes of *weak baseline detector*, combine it with semantic similarities, and select top-k classes ranked on comprehensive similarities. Visual similarity is presently the difference between the weights of the last layers of seen and unseen classes.

4. I transfer the knowledge of classifier-detector differences of top classes to the constructed unseen class detector and adapt it into the stronger detector without further training.

5. Finally, I perform the detection using our trained network of YOLO-MobileNet and communicate results.

We describe below the construction of our strong and weak baseline detectors offline for *seen* concepts and training of detectors online for *unseen* concepts while investigating the object detection model's training time, which we refer to as the *response-time* of our model on *unseen* concepts. Since LSDA established the background of conversion of image classifiers into object detectors, we are using its guidelines to construct our model while using MobileNetv3 (Small) [151, 306] with YOLOv3 [35, 166] in-place of AlextNet [305] and R-CNN [307]. In our design, we assume classifier to detector conversion methodology eliminates the need for bounding boxes and the use of visually/semantically similar classifiers for the knowledge transfer speeds up the training and proves this Hypothesis-IV in the next section.

8.4.2.1 Training Baseline Detector Offline for Seen Concept (with Bounded Vocabulary)

First, we set up an architecture of YOLO with MobileNet backbone and construct two baseline detectors as follows:

Strong Baseline Detector (D_S) is a $|K|$ class detector trained on existing object detection datasets. It is a detector that is trained on strong labels (i.e., bounding box annotations). Presently we have taken 100 classes (like LSDA) by considering all classes of Microsoft COCO (80 classes [27]) and 20 classes of OID [28]. Please note that 20 classes of Pascal VOC [26] are also present in Microsoft COCO.

Weak Baseline Detector (D_W) is another $|K|$ class detector trained on image classification dataset. We trained it on weak labels (i.e., images-level labels). In this detector, we consider the same classes on which we trained the previous *Strong Baseline Detector*, but we use the ILSVRC [152] classification data. The value of $|K|$ is 100 in both cases.

8.4.2.2 Training Online Detector for Unseen Concept (for Unbounded Vocabulary)

On request of an unseen class (u), say *goat*, first our model provides an environment to collect images for 'goat" from the Web using Google Images[1], Flickr[2], or Bing Image[3] search. Second, it uses the "Strong Baseline Detector" and sets up a new detector by labeling the most similar *seen* class (like sheep) with the *unseen* class (i.e., goat). In other words, it renames the seen class (sheep) with the unseen class (goat). It is important to note that a similar class (like sheep for goat) can be chosen only using semantic similarity at this stage as visual features of an unseen class cannot be computed before training. Next, we fine-tune the detector on images collected for "goat". Now we have a new detector having $|K|$ classes that can detect *goat*. Since *goat* class is trained only on image-level labels (weak labels), we call it a *weak detector* or merely a classifier "C_u" for unseen class. At this point, our model's response time for *unseen* concepts is equal to the time for fine-tuning.

We presume that fine-tuning induces a *specific category* bias transformation in the detection network towards class "goat" (which is complimentary from the viewpoint of detecting a class goat). Moreover, this network already encodes a *generic "background" category* having been previously trained on detection data (because of strong baseline), which is another positive perspective, as this will automatically make the new detector much more effective in localizing the new class without detection data. Finally, the previous classifier C_u adapts into a corresponding detector D_u using the same assumption "difference between classification and detection of a target object category has a positive correlation with similar categories" of LSDA. Suppose weights of the output layer of D_S (Strong Baseline Detector) and D_W (Weak Baseline Detector) are w^{D_S} and w^{D_W}, respectively. We know that for any *seen* category $i \in K$, final detection weights should be computed as $w_i^{D_S} = w_i^{D_W} + \delta_{K_i}$, where δ_{K_i} is the difference $(w_i^{D_S} - w_i^{D_W})$ in weights of output layer of the seen category (Strong and Weak Baseline) detectors.

[1]https://github.com/hardikvasa/google-images-download
[2]https://www.flickr.com/services/api/
[3]https://pypi.org/project/bing-image-downloader/

By using this knowledge difference and denoting the k^{th} nearest neighbor in set K of category u as $N_K(u,k)$, we adapt the final output detection weights for categories u as:

$$w_u^{D_u} = w_u^{C_u} + \sum_{i=1}^{k} s(u,i)\delta K_{N_K(u,i)} \tag{8.2}$$

where $k \leq |K|$, and $s(u,i)$ denotes the similarity of seen class (i) with unseen class (u).

The main difference between Eq.8.1 and 8.2 is the *weighted nearest neighbor* scheme [30, 61], where weights are assigned to seen categories based on how similar they are to the unseen category. We select top-k weighted nearest neighbor categories $(s(u,i))$ using Eq.8.3. Besides the semantic similarity, we also compute the visual similarity at this stage by using the minimal Euclidean distance between the detection parameters of the last layers of detectors D_W and C_u. Suppose K_v is the set of visually similar (s_v) categories and K_s is the set of semantically similar (s_s) categories, then comprehensive similarity $s(u,i)$ for the unseen category with seen categories is evaluated as:

$$s(u,i) = \alpha s_v(u,i) + (1-\alpha)s_s(u,i),\ \ i \in \{K_v \cap K_s\} \tag{8.3}$$

where $\alpha \in [0,1]$ is a parameter introduced in [30, 61] to control the influence of the two similarity measures. However, Tang et al. [30] proposed a modified visual similarity model and used that in LSDA with the *weighted average scheme* while leaving the impact of the weighted average scheme over the simplified visual similarity measure used in LSDA. With the improved visual similarity model, Tang et al. [30] model focuses only on increasing accuracy, unlike UnseenNet, which focuses on reducing training time.

Since the aim of our model is to reduce the overall response time, we use the LSDA based visual similarity [66] and naive *path-based* semantic similarity measure of WordNet [29] along with a weighted average scheme to compute the comprehensive similarity $(s(u,i))$ scores. We verify the value $\alpha = 0.6$ on simplified similarity measures by analyzing the performance (shown in Section–8.4.2.3).

Finally, we call this adapted detector "D_u", a *strong detector* for unseen class. We analyze the response-time of our model in Section–8.5 from the stage of *no detector* to *weak detector (C_u)*, and eventually to a *strong detector (D_u)*.

8.4.2.3 Implementation Details

Data Preparation I use 100 seen and 100 unseen Classes throughout the experiments. I trained *Strong* and *Weak Baseline Detectors* on 100 seen Classes offline and performed experiments on 100 unseen classes while having training time constraints.

- **Seen Classes**

  Strong Baseline Detector Training: In this case, I consider all 80 classes of Microsoft COCO [27] and 20 classes of OID [28] to train a strong baseline detector with bounding box annotations. I select 20 classes from OID by sorting its 600 classes on the basis number of images per class and considering the top 20 with the highest number of images available for training.

  Weak Baseline Detector Training: Here, I take the same 100 seen classes, retrieve images with labels from the ISLVRC [152] dataset (i.e., images have no bounding boxes), and train *weak baseline detector* by giving full image size in place of annotations.

- **Unseen Classes**

  Training: Similar to LSDA [3], I take another 100 classes from the ILSVRC [152] to train unseen classes.

  Testing: I chose these 100 unseen classes in such a way that the same classes should be present in OID (consist of 600 classes). I evaluate the model on an object detection dataset, which gets trained on image classification dataset. That is, I use the testing dataset of OID for 100 unseen classes to serve as groundtruth in the evaluations.

I also show qualitative evaluations on additional 16 unseen classes that I downloaded from the web using Google Images API [4]. Such classes are not present in any dataset to-date and prove model's significance for unseen concepts (known or unknown).

Training In my experiments, I consider the main settings of LSDA while using the pipeline of YOLOv3 [35, 166] and MobileNetv3 [151, 306]. Specifically, I used the three layers (38, 117, 165) from the MobileNetv3 (Small) within YOLO to make the prediction[5].

I trained the baseline detectors first on learning rate of 10^{-3} till 100 epochs, then I used the decay type exponential till 200 epochs; finally, I used the 10^{-4} till 300 epochs as validation loss stopped decreasing near this point. However, for the training of the unseen classes, I used the constant learning rate of 10^{-4}, which could be increased in future experiments for faster results. I kept the slowest possible learning rate, as my model should serve as the base-work for handling dynamic unseen concepts in short training time. Finally, I utilize the benchmark object detection metrics project [6] to evaluate the detections with IOU=0.5.

[4] https://github.com/hardikvasa/google-images-download
[5] https://github.com/david8862/keras-YOLOv3-model-set
[6] https://github.com/Adamdad/Object-Detection-Metrics

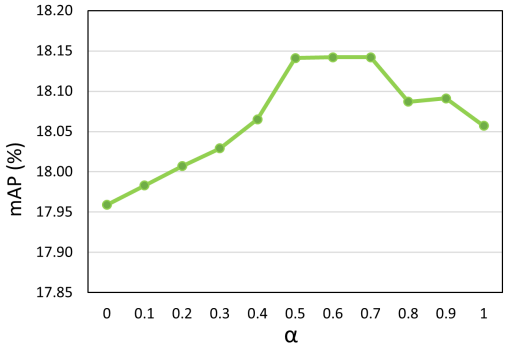


FIGURE 8.5: mAP with parameter α for degree of similarity.

I assume it is essential to specify that ImageNet and Object detection datasets use different name for the same classes, so I am using the vocabulary of WordNet to give a single name to each class and also provide mappings of different datasets with our model.

I used the path vector of WordNet for the semantic similarity measure. Visual similarity is simply computed using the minimal Euclidean distance of weights of the unseen class detector (trained on classification data) and weights of weak baseline detector, which is the same as described in LSDA. Here I use a degree of similarity measure to compute the comprehensive similarity between seen and unseen classes.

Degree of Similarity Parameter (α) To complete the weighted average scheme's evaluations over simplified (visual and semantic) similarity measures, I analyze the value of parameter α. Figure 8.5 shows impact of α on mAP; its peak values could be 0.5, 0.6, and 0.7.

Estimation of Number of Epochs I estimate the total number of epochs required to train the model for the designated training time (minimum, mean, or maximum) by considering the batch size, total number of training images available for a particular class, and speed of our GPU for the completion of one step. The total number of epochs computed as:

$$epochs = \frac{ResponseTime}{((Num\ of\ Images/Batch\ Size) * t)} \tag{8.4}$$

where, "response time" denotes the total training time allowed, "Num of Images" is the number of available training images, and t is time GPU takes to complete one step, which is 0.465 sec in our case. Here, "Num of Images/Batch-Size" is the number of steps. We used default batch-size 16. We conducted experiments on NVIDIA TITAN Xp GPU (8 Core Processor×16), Driver 440.1 with CUDA 10.2.

8.5 Evaluation

This section describes the evaluation methodology, evaluation metrics, and finally, experiments with results.

8.5.1 Evaluation Methodology

The evaluations present in the work are divided into two broad categories: *Quantitative* and *Qualitative Evaluation on Unseen Categories*. Quantitative evaluations are further classified into three categories. First, we compare our model performance with existing models to validate our Hypothesis-IV. Second, we show our model's performance with a timeline of response-time, and third, we present results on 100 unseen concepts with their degree of similarities with unseen concepts. Our qualitative evaluations show visual examples of correct and incorrect detections of "UnseenNet". These qualitative evaluations include additional 16 unseen classes that are not present in any dataset to-date to prove our model's significance on unseen concepts.

8.5.2 Evaluation Metrics

- **Response-Time**: The training time of object detection models in responding to unseen concepts contributes towards the *response-time*.
- **mean Average Precision (mAP)**: The *mAP* is the average of the average precision of all classes. It is computed by calculating AP separately for each class, then average over them.

8.5.3 Experiments and Results

8.5.3.1 Quantitative Evaluation on Unseen Categories

Comparative Analysis with Existing Models We compare the performance of the UnseenNet in Table 8.2 against LSDA and semi-supervised LSDA. We show mean average precision (mAP) for unseen categories along with required training time. We evaluate our model by considering different number (5, 10, and 100) of nearest neighbors of "unseen" categories with "seen" categories while using the weighted average nearest neighbor scheme (Eq 8.2), same as the other LSDA based methods [30, 61].

The first 5 rows show the baseline results of LSDA. We also include the performance of semi-supervised LSDA. The last row shows the detection results of an oracle detection

network, which assumes that bounding boxes for all 100 "unseen" categories are available and no constraint on training time. The first row shows the detection results by training the network only on classification data without adaptation. Then the next rows show class invariant and class-specific adaptation results of baseline LSDA. Also, we include results in the 6^{th} Row of a semi-supervised version of LSDA [30, 61]. We observe that training time is very high (>5.5 hours) in existing scenarios. Also, LSDA settings inference time is 2 fps, and UnseenNet is 9.2 fps because of using YOLO and MobileNet.

It is necessary to evaluate our model first by training only on classification data because we are using YOLOv3–MobileNetv3 [166, 306] in contrast to R-CNN–AlexNet [305, 307]. We show that this amendment improves the performance from 10.31 to 12.79 with a large decrease in response time from $\sim$ 5.5 hours to $\leq$ 10 min. Here we show the mAP for different response times (5 min, 10 min, 20 $\rightarrow$ 50 min). We choose these response times using the results of testing and training (shown in Figure 8.6) detail in Section–8.5.3.1.

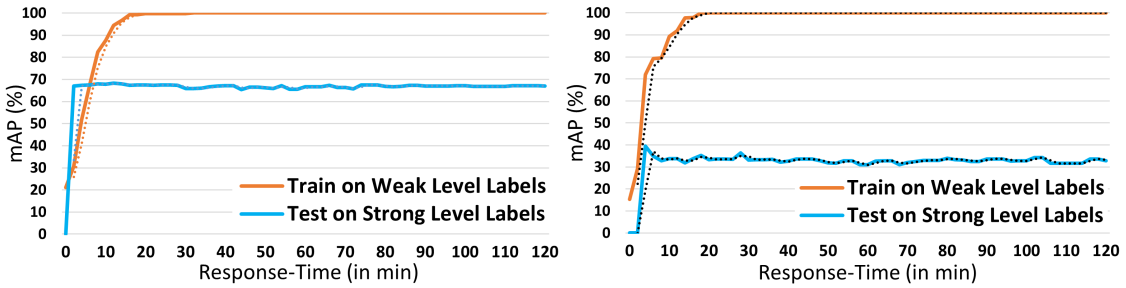
Second, we show the mAP using Class Invariant Adapt (Strong Baseline Detector) and fine-tuning the nearest "seen" class on target "unseen" class classification data. Finally, we apply the specific class adaptation using the weighted average of "N" nearest neighbor classes, where N could be 5, 10, and 100. This step does not require training. We show the final detection performance (average on 100 classes) by indicating our model's total time.

Best results indicate that we can reach from stage of **no** detector for unseen concepts to a weak detector (mAP **18.96**) and strong detector (mAP **19.82**) within **5 min** of training. These results validate our Hypothesis-IV as a conventional classifier to detector conversion methods (LSDA and improved LSDA) reaches to mAP of 16:33 and 20:03 in >5.5 hours of training, where visually/semantically knowledge transfer is not utilized for initiating the training. Moreover, the oracle network, which needs reaches to the mAP 28:59 while taking >120 hours (not appliable for real-time training-based applications).

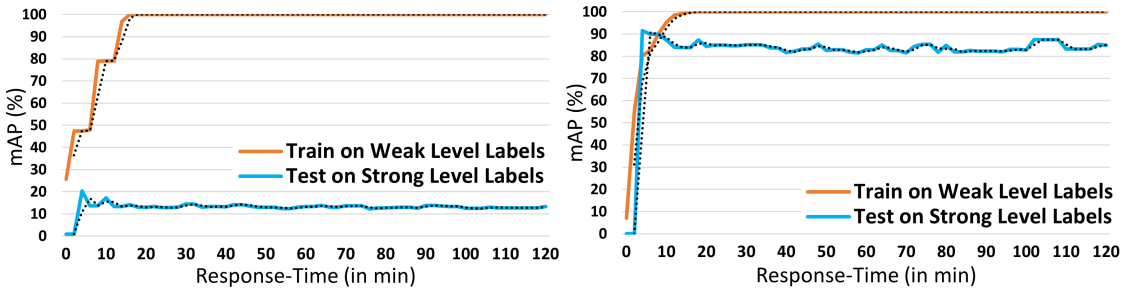
Experimental Results with Response-Time To retrieve the effective range of response-time in our model, we train each category until the point testing accuracy starts to decrease (to avoid overfitting). We show a few examples of unseen concepts in Figure 8.6. Please note here we compute the total number of epochs for varying the training time (detail in Section–8.4.2.3). We first train our model on weak level labels (i.e., without bounding boxes) and then test on strong labels (i.e., with bounding boxes). We observe that the maximum mAP of each class could be achieved within 10 min of training. After that, the mAP decreases and remains constant. Thus we suggest 20 min of training as the maximum time limit. However, we recommend 5 min of training to attain maximum mAP 19.82 and 10 min to avoid any unexpected reduction in mAP due

Method		**Number of Nearest Neighbors in "Seen" categories**	**mAP on "Unseen" 100 Categories**	
			mAP	**Response-Time**
LSDA (**Baseline**)	(Classification Network with No Adapt)	–	10.31	5.5 *hours*
	(Only class invariant adaptation)	–	15.85	
	(Class Invariant & Specific Adapt)	Weighted Avg NN - 5	16.12	
		Weighted Avg NN - 10	16.28	
		Weighted Avg NN - 100	16.33	
Semi-Supervised LSDA	(Incorporating Visual and Semantic Knowledge)	–	20.03	$>$ 5.5 *hours (LSDA settings + Informed Visual Transfer)*
UnseenNet	(Classification Network with No Adapt)	–	12.79	5 *min*
		–	13.45	10 *min*
		–	17.81→16.46	20 → 50 *min*
	(Class Invariant Adapt & Specific Class Fine-Tuning)	–	18.96	5 *min*
		–	17.74	10 *min*
		–	17.07→16.81	20 → 50 *min*
	(Class Invariant Adapt, Specific Class Fine-Tuning & Adapt)	Weighted Avg NN - 5	**19.09**	**5 min**
			17.80	10 *min*
			17.10→16.84	20 → 50 *min*
		Weighted Avg NN - 10	**19.21**	**5 min**
			17.88	10 *min*
			17.13→16.86	20 → 50 *min*
		Weighted Avg NN - 100	**19.82**	**5 min**
			18.14	10 *min*
			17.28→16.94	20 → 50 *min*
Oracle	Full Detection Network		28.59	$>$ 120 *hours*

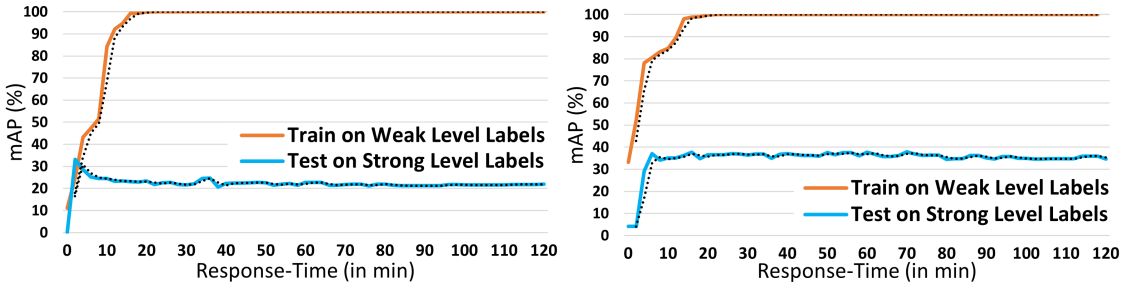
TABLE 8.2: The mean average precision (mAP) while using ILSVRC for Weak Level labels and Microsoft COCO & OID for Strong Level labels.



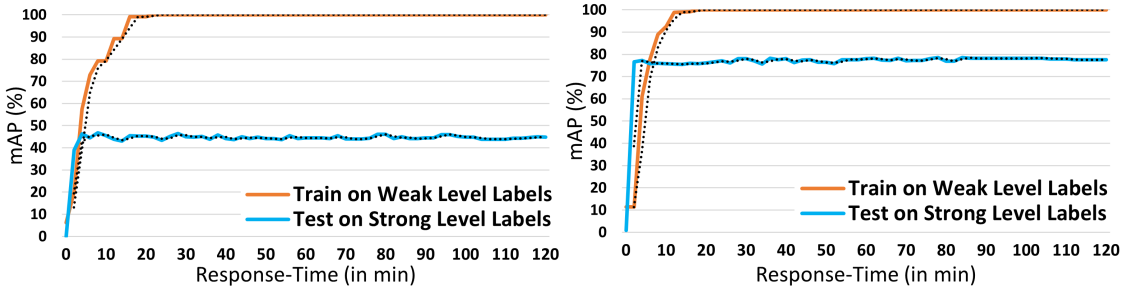
(a) Building (b) Cake



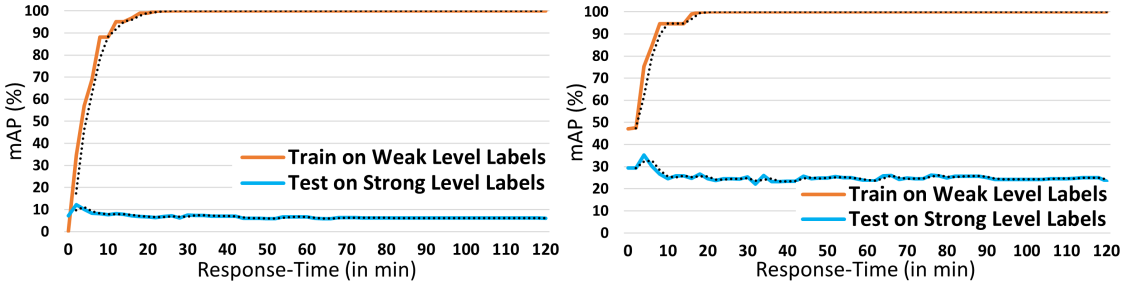
(c) Mushroom (d) Pasta



(e) Goat (f) Candy



(g) Swimming_pool (h) Poster



(i) Dress (j) Spider

FIGURE 8.6: Examples of mAP with Response-Time, For each "Unseen" category, we use the top-10 weighted average nearest neighbor "Seen" categories for adaptation.

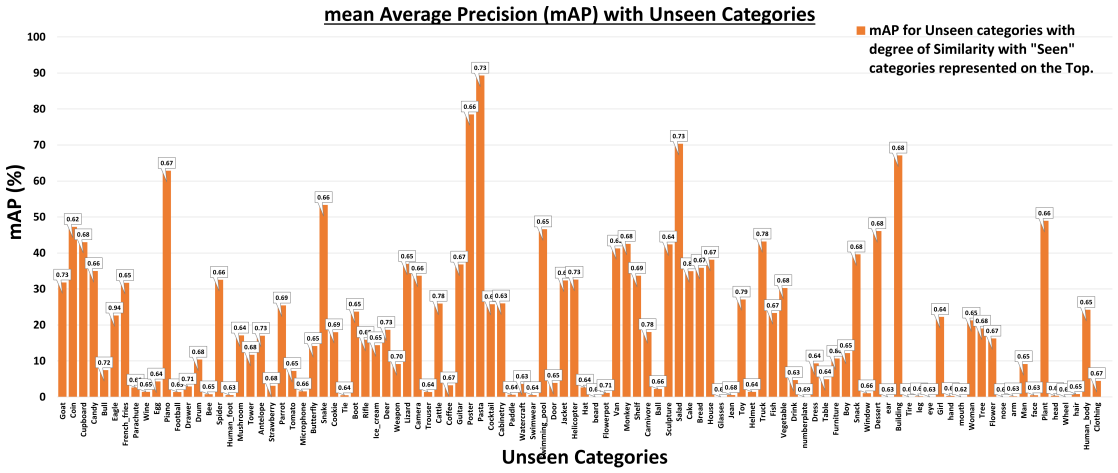


FIGURE 8.7: mAP of our model on 100 "Unseen" Categories within 5 min of training.

to unstable training at the initial stages. It is worth noting that mAP at 0 min training time is not zero due to the strong baseline detector for initialization.

Experimental Results with Unseen Concepts We present an analysis in Figure 8.7 of 100 *unseen* categories along with their respective degree of similarity with *seen* categories (top-10 nearest neighbor). The simple average similarity score is:

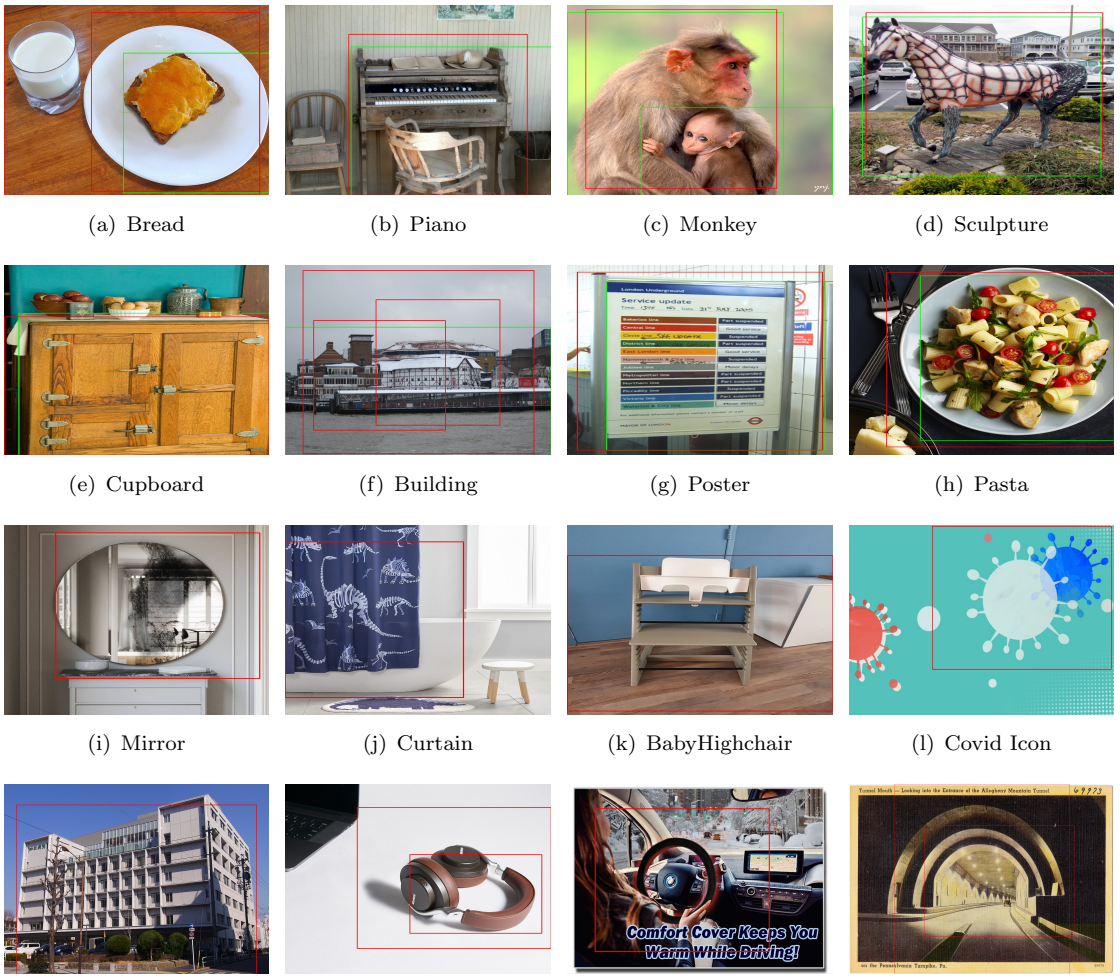
$$s_j = \frac{\sum_{i=1}^{m} s(j,i)}{m} \tag{8.5}$$

where m is 10 presently and $s(j,i)$ is the comprehensive similarity (shown in Eq. 8.3) between unseen (j) and seen (i) category computed using α=0.6. It shows if we have *unseen* classes (like building, pasta, salad) more similar to *seen* classes, then our model gives high performance with the exception for small size objects (like microphone, strawberry) or availability of less training data (cattle, jeans).

8.5.3.2 Qualitative Evaluation on Unseen Categories

We show visual examples of our model detections in Figure 8.8. Here Figure 8.8 (a) – (h) includes classes of ILSVRC, and Figure 8.8 (i) – (p) consists of additional unseen classes which not present in any object detection or image classification dataset to date. Correct detections of unseen concepts verify that UnseenNet can be trained on any class within 5 min of training. It also reduces the need to create large object detection datasets.

Some examples of incorrect detections are shown in Figure 8.9. This demonstrates that if we have "unseen" classes less similar to "seen" classes, then UnseenNet could label



(a) Bread (b) Piano (c) Monkey (d) Sculpture

(e) Cupboard (f) Building (g) Poster (h) Pasta

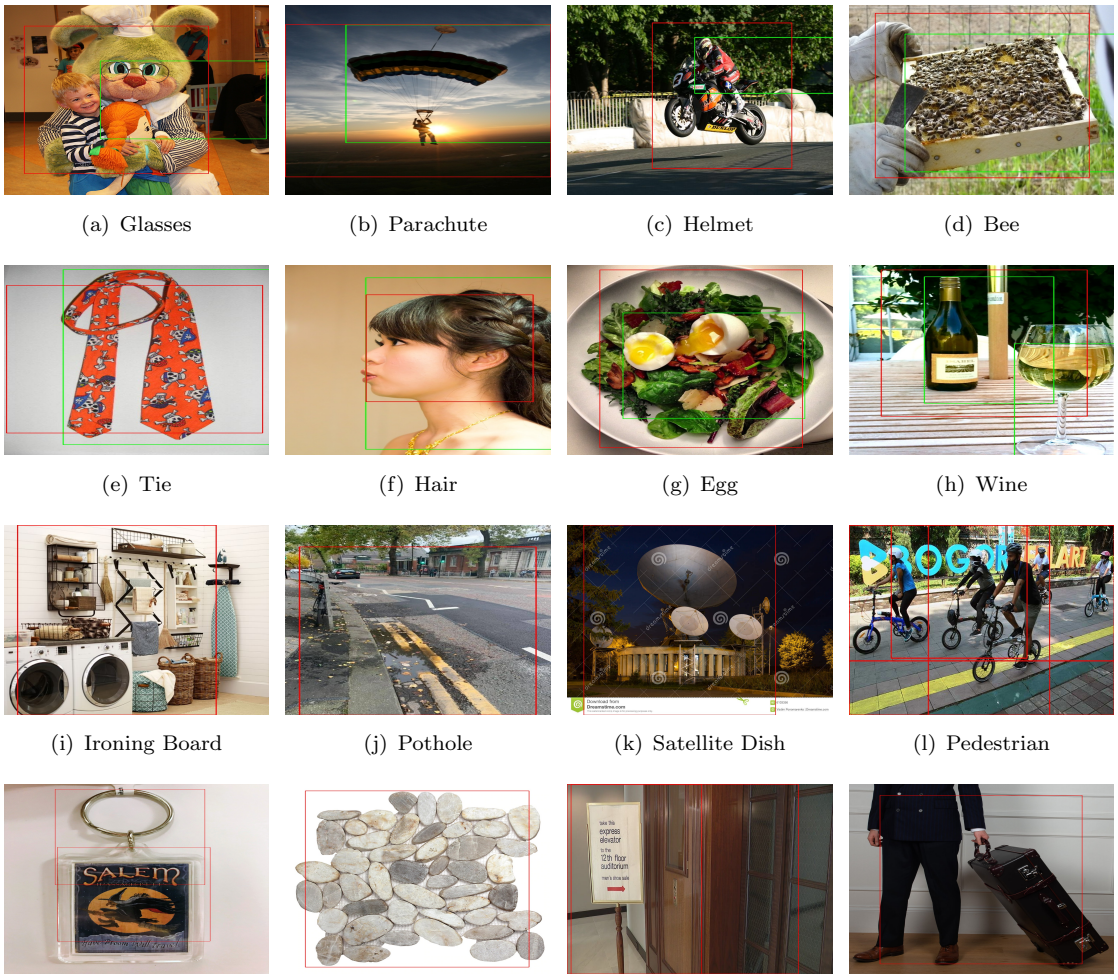
(i) Mirror (j) Curtain (k) BabyHighchair (l) Covid Icon

(m) Hospital (n) Headphones (o) Steering Wheel (p) TunnelEntrance

FIGURE 8.8: Examples of correct detections of our model on "Unseen" categories are shown in red color and groundtruth (taken from OID) in green. Last two row unseen classes are downloaded online, and no groundtruth available to date.

them *correctly* because of the training on classification data with *incorrect* localization due to the absence of detection data.

Besides the observations and results presented in this section, I assume increasing the learning rate (taken as 10^{-4}) could reduce the response time further but may or may not cost accuracy. Evaluations of Model-IV for the domain where more seen classes are available can increase the accuracy. I also believe testing of UnseenNet is feasible on edge devices; however, response time may increase. Moreover, there will be a need to include classifier transfer time due to the construction of more classifiers (each 100MB) depending on unseen subscriptions.



(a) Glasses (b) Parachute (c) Helmet (d) Bee

(e) Tie (f) Hair (g) Egg (h) Wine

(i) Ironing Board (j) Pothole (k) Satellite Dish (l) Pedestrian

(m) Key (n) Stone (o) Elevator (p) handle

FIGURE 8.9: Examples of Incorrect detections (Label Object Correctly but Incorrect Localization) are shown in red and groundtruth (taken from OID) in green. Last two row unseen classes are downloaded online, and no groundtruth available to date.

8.6 Conclusion and Discussion

In this chapter, I presented an "UnseenNet" model that has the ability to construct a detector for any unseen concept without bounding boxes while training in a short time and providing competitive accuracy. I found that starting from a "strong baseline detector" trained on existing object detection datasets speeds up the training rather than using only the ImageNet [152] pre-trained model to train unseen concepts. Moreover, in conjunction with semantic and visual similarity measures, classifier-detector conversion methods make our approach achieve comparable mAP 19.82. The evaluations demonstrate that *UnseenNet* outperforms the baseline approaches and reduce the training time of days or > 5.5 hours to < 5 min. Hypothesis IV about the speeding up of training with no need for bounding boxes has been clearly validated due to the considerable reduction in response-time and competitive performance using only image-level labels.

A limitation/observation of the proposed "UnseenNet" is that it constructs a new detector for each class; Is this how not wasting space? Assuming the average size of the detector is 100MB, and we get 1 million unseen concepts (which is already highly unlikely). Then total space detectors will occupy $=100\text{MB}\times10^6=100\text{TB}$, which is not too large space in the era of having 1TB hard disk on personal laptops when we can handle millions of classes. In the future, UnseenNet could be improved with more effective detectors and classifiers other than YOLOv3 and MobileNetv3. Strong and Weak baseline detectors could include a large number of *seen* classes to obtain more similar classes. Investigating few-shot learning [63, 167–170] is also a reasonable future direction. I detail the summary of conclusions with possible future solutions in the next Chapter-9.

Chapter 9

Conclusion and Future Work

9.1 Thesis Summary

Explosive growth in the number of physical devices being connects to the Internet observed in recent years. The impact of this increase also provides clear evidence of shifting the global network towards internet traffic of multimedia. For example, events related to traffic congestion, accidents, change in weather, parking problems, security, pedestrian detection, etc., belong to multimedia (unstructured) events. This enormous generation of multimedia data (i.e., images, video, and audio) within smart cities compelled us to move from conventional IoT to IoMT (Internet of Multimedia Things). Event-based approaches in IoT are mainly efficient in processing structured (scalar) events of smart cities and have a limited focus on IoMT. Advancements in Deep Neural Network (DNN) may support IoMT data but require the availability of trained classifiers for unseen (new) concepts. The limitation of having to train classifiers for unseen concepts may increase the overall response-time for multimedia-based event processing models. This work focuses on the problem of multimedia event processing, which includes redefining event processing to *multimedia event processing*, introducing *detection operator* for event query languages, standardizing the concept of *response-time*, proposed multiple IoMT based *deep neural network models* for object detection specifically, and established a fast *online training detector* for *unseen* concepts *without bounding box annotations*.

Chapter–2 focused on problem formulation, requirements, challenges, and motivation of multimedia event processing for IoMT. The problem domain also emphasizes the limitations of online training of classifiers for "unseen" concepts and the availability of the type of training data with/without bounding boxes. Suppose a user subscribes for *pedestrian* class detection, and the existing public traffic control management system can recognize only *bus*, *taxi*, *traffic-light*, *etc.* In that case, the system may require manual

effort to answer any unseen subscription like "pedestrian". However, with the provision of online training, such types of queries can be answered automatically by training a new *pedestrian* classifier in a short time. In this chapter, I also formulate the concept of "response-time" that I used throughout our models, using the test cases of online training based on presence or absence of classifier, type of training (scratch or domain adaptation), and kind of training data (with or without bounding boxes). Lastly, I divide problem "*How can we answer user queries online consisting of seen (bounded vocabulary) as well as unseen subscriptions (unbounded vocabulary) that include processing of multimedia events while achieving high accuracy and minimizing the response-time, where the training of classifiers may or may not have bounding box annotations available?*" in specific research questions addressed by proposed models.

Chapter–3 analyzed state of art for IoMT based systems and presented visions of IoMT in light of IoT. Efficient deep neural network-based object detection models could prove to be an asset for training unseen concepts in IOMT. Background of object detection and fully annotated datasets with their comparative analysis is also provided in this chapter. Due to the limited vocabulary of object detection datasets, existing deep neural network-based models cannot be used to train large vocabulary models.

Chapter–4 first described the generalizable multimedia event processing using event processing. Next, I detailed its scenarios adhering to seen/unseen concept problem identified using three conditions, namely "Is the concept Unseen?", "Any similar Seen concept available?", and "Are Bounding Boxes inaccessible?". Finally, the rationale for proposed models using their associated scenarios is discussed along with their contributions that mainly lie in optimizing online testing (for Model I) and online training (for Model II, III, and IV) time to reduce the overall response-time for multimedia event processing.

Among different presented scenarios, the first model (Model-I, Chapter–5) analyzes the foremost basic scenario related to the arrival of seen (i.e., familiar) subscriptions. Here, the main contribution includes a multimedia stream processing engine with a neural network-based event matcher using a "detect" operator and an optimization technique focused on reducing the testing (inference) time. In the "classifier division and selection" based optimization approach, Model-I selects only domain-specific classifiers to process subscriptions. For instance, the "car" classifier (single class classifier) will be chosen by the proposed model to detect a car. The associated publication to this chapter is [32].

Chapter–6 is focused on the scenario of completely unseen subscriptions. Here, the main contribution consists of an adaptive architecture for multimedia event processing and response-time based strategies with their respective prototypes by tuning hyperparameters for the optimized training. Since the choice of hyperparameter values dramatically affects the performance of resulting classifiers, I leverage hyperparameter tuning based

techniques that include the configuration of learning-rate, batch-size, and the number of epochs for minimizing the response time. The associated publication to this chapter is [33].

Chapter–7 covers the scenarios of unseen subscriptions where domain adaptation is feasible. Here, I provided the evident solution of reducing response-time by introducing the notion of adaptation among classifiers (either inter or intradomain) for partial unseen concepts. In this work, I mainly instantiated the online classifier learning model by transferring knowledge among classifiers using fine-tuning and freezing layers of object detection models. The associated publication to this chapter is [34]; a paper titled "Detecting Seen/Unseen Concepts while Reducing Response Time using Domain Transfer in Multimedia Event Processing" is under submission in the IEEE Access.

Chapter–8 investigated the last scenario where bounding box annotations maybe not be available to train object detection models on unseen concepts. In this work, I proposed an "Unseen Detector" that can be trained within a short time for any unseen class without bounding boxes with competitive accuracy. Unseen concepts are fine-tuned on the strong baseline detector using only image-level labels and further adapted by transferring the classifier-detector knowledge between baselines. We use semantic and visual similarities to identify the source class (i.e., Sheep) for the fine-tuning and adaptation of unseen class (i.e., Goat). The associated publication to this chapter is [38], and a paper titled "UnseenNet: LSDA-based Fast Training Detector for Unseen Concepts with No Bounding Boxes" is under submission in IEEE TPAMI.

The summary of conclusions derived by extensive experiments of our proposed models is discussed in the below section.

9.2 Conclusions

In this work, an adaptive approach for multimedia event processing has been proposed, using domain knowledge transfers while online classifier construction of object detection models to handle unseen subscriptions in low response-time. The proposed model has been optimized at various stages using classifier division and selection, tuning of hyper-parameters, and transfer of domains based techniques with and without bounding boxes. The performance is enhanced in terms of the accuracy of processing unseen subscriptions with the reduction in response-time in each model step. We describe below the derived conclusions from each model along with their associated hypotheses.

Hypothesis-I: If we construct N-Class classifiers for different domains, and we use subscription constraints to choose closely related classifier for the

processing of multimedia events; the performance will get enhanced in terms of accuracy and response time, and will also add the ability to generalize for multiple domains.

The first model (Chapter–5) related to domain-specific classifier based multimedia event processing interprets this research hypothesis. I performed experiments on three types of classifiers: single class, N-class, and 80-class classifiers to validate this hypothesis. The 80-class classifier serves the purpose of a general classifier having all 80 classes of Microsoft COCO consisting of multiple domain categories. Single-class classifiers are constructed the same way as binary classifiers, predicting whether a particular class is present in an image. For N-class classifiers, we prepared events (using Microsoft COCO) sets consisting of events related to four separate domains, i.e., traffic, sports, home, animals, and one mixed domains event stream. Here the value of "N" depends on the application domain, which is taken as 8, 9, 17, 10, and 44 for traffic, sports, home, animal, and mixed classifier, respectively.

The evaluations found that the domain-specific (N-Class) classifiers consistently outperform the other classifiers (general purpose and single-class classifiers) with an average throughput and accuracy of 110 fps and 66.34%, respectively. The high average throughput of N-Class classifiers on event streams for the concepts presents in classifiers and achieving high accuracy on different domains signifies the generalizability, efficiency, and effectiveness of the proposed model for real-time applications. Further, we varied the value of N from 1 to 80 (i.e., 1-class classifier, 2-class classifier, 3-class classifier, and so on); and I demonstrated the system's performance would decrease from 115fps to 107fps with an increase in the number of classes per classifier. In addition to the accuracy, I also verified the high precision and recall of N-class classifiers to prove optimization.

Hypothesis-II: If tuning of hyperparameters based technique is useful in machine learning models to speed-up the training, decrease the computation cost, and increase the accuracy; then performance will get enhanced for low response-time also even on training from scratch for unseen subscriptions on tuning hyperparameters for the online construction of classifiers.

Model-II (Chapter–6) based on hyper-parameter based adaptive multimedia event detection validate this research hypothesis. I verified the hypothesis by performing experiments before and after adaptation (i.e., without and with the tuning of hyperparameters) on completely unseen concepts. First, I analyzed the trade-off between response time and performance (mAP) using default hyper-parameters on object detection models YOLO, SSD, and RetinaNet [35–37]. Then I identified and changed the configuration with hyperparameters to adapt the object detection models for low response time.

I observed that the accuracy of each model before adaptation increases after adaptation for two strategies S1: Minimum Response Time needed while Minimum Accuracy allowed and S2: Optimal Response Time needed while Optimal Accuracy allowed. Specifically, it increased from 0.00% to 5.66%, 10.08% to 47.32%, and 64.66% to 79.00% for YOLO, SSD, and RetinaNet, respectively for S1. Correspondingly, for S2, the accuracy of YOLO increased from 79.16% to 82.82%, SSD slightly changed from 54.79% to 54.81%, and RetinaNet considerably increased from 74.87% to 84.28%. Please note in strategy S1, I considered the training time of 15 min, and S2 covers the training time of 60 min. The enhancement in performance on such low training time of object detection models on the tuning of hyperparameters proves the proposed hypothesis of online construction of classifiers in low response-time.

Hypothesis-III: If transferring of knowledge from one domain to another (say $A \rightarrow B$) can improve the performance as compared to fine-tuning of pre-trained models (like $C_{P_{ImageNet} \rightarrow B}$) or training of classifier from scratch (C_B); then there will always be a decrease in response-time with increase in accuracy of constructed classifier ($C_{A \rightarrow B}$) than the classifier trained from pretrained model (like $C_{P_{ImageNet} \rightarrow B}$) or training from scratch (C_B).

Model-III (Chapter–7) designed for domain adaptation based multimedia event detection model give proof of this research hypothesis. To prove the hypothesis, I compared object detection models using mean average precision (mAP) and response time as performance metrics. We trained models on three training techniques (i.e., training from scratch, fine-tuning of pre-trained models (like $C_{P_{ImageNet} \rightarrow B}$), and direct domain transfer (from $A \rightarrow B$)) for 120 min. I observed that in all object detection models, i.e., YOLOv3, SSD, and RetinaNet, I get the mAP of 0.1, 0.12, and 0.16, respectively, on direct domain transfer within a response-time of 0min which strongly supports our hypothesis. Since we aim to compute the best performance while minimizing response-time, I choose YOLOv3 as it performs better among all models at most of the short training time. I found that for the short training time (like 30min), YOLOv3 with freezing technique (mAP $\simeq 0.50$) performs best while its fine-tuning counterpart achieves mAP $\simeq 0.11$, which also supports our hypothesis.

Hypothesis-IV: If an adaptation of classifier into detector eliminates the need of bounding boxes as well as transferring of knowledge from one domain to another speed-up the training; and a detector gets constructed from classifier with the help of transfer of knowledge from visually/semantically similar classifier; then that detector will take less time to train for unseen classes and eliminate the requirement of bounding boxes.

Model-IV (Chapter–8) focused on domain adaptation without bounding boxes testify this research hypothesis. The hypothesis has been validated through experiments of the proposed "UnseenNet" model. I consider 100 seen classes (present in Pascal VOC, Microsoft COCO, and OpenImages dataset) and 100 unseen classes (present in ILSVRC) for evaluation. I compared the performance of the UnseenNet against LSDA, semi-supervised LSDA, and an oracle detection network. Here, the oracle detection network assumes that bounding boxes for all "unseen" categories are available. However, the LSDA model uses the classifier-to-detector conversion method to eliminate the need for bounding boxes while using visual/semantic knowledge transfer after training and not during the training. Semi-Supervised also follows the same approach as LSDA while using improved versions of visual similarities. Contrarily, UnseenNet uses a classifier to detector conversion method, train while transferring knowledge from one domain to another (i.e., from seen to unseen classes), and adapt after training. Due to this reason, LSDA reached mAP 16.33 in 5.5 hours of training, and improved LSDA achieved mAP 20.03 in $>$ 5.5 hours. However, the UnseenNet reached the mAP 19.82 within only 5 min of training without the need to bounding box annotations. Moreover, the oracle network, which gets trained on bounding boxes, reached the mAP 28.59 while taking $>$ 120 hours. These results state the advantage of using transfer learning for the fast training of unseen class classifiers and the use of classifier to detector methods to eliminate the need for bounding boxes, verifying the hypothesis.

9.3 Core Contributions

The contributions of this work can be summarized as follows:

Problem Formulation for the Multimedia Event Processing using Online Classifier Training: We formulated the problem of processing multimedia events for dynamic subscriptions (concepts) using domain-specific classifiers, online training, and transfer learning-based large-scale domain adaptation approaches for covering the requirement of *generalizability* and supporting *seen/unseen subscriptions* in reduced *response-time*.

Neural Network based Matcher with "DETECT" Operator: We proposed a neural network-based event matcher for processing multimedia events and optimized it using subscription constraints, with the provision of a "detect" operator in event query languages to support expressly object detection.

Standardization of objective function "Response-Time": We standardized the objective function "Response-Time" for the adaptive multimedia event processing and provided response-time based strategies with their respective prototypes by tuning *hyperparameters* for the real-time classifier training.

Adaptive Framework for Online Classifier Construction: We presented an adaptive architecture for online classifier construction to minimize the *response-time* and maximize the *accuracy*.

Instantiation of Online Classifier Learning model using Fine-tuning & Freezing Neural-Network Layers: We instantiated the adaptive online classifier learning model by transferring knowledge among classifiers using fine-tuning and freezing layers of neural network-based object detection models.

Evaluation of Proposed Models using Object Detection methods with Response-Time & Accuracy: We enhanced the performance of object detection models (YOLO, SSD, and RetinaNet [35–37]) for processing multimedia events on dynamic (seen/unseen) concepts belonging to Pascal VOC, Microsoft COCO, and OpenImages datasets [26–28], which achieved:

- an accuracy of 66.34% with permissible response-time of 2-hours in *domain-specific classifier based multimedia event detection* approach.
- an accuracy of 84.28% within 1-hour response-time by using *hyperparameter tuning based multimedia event detection* approach.
- an accuracy of 95.14% within 30-min of response-time while using *domain adaptation based multimedia event detection* approach.

UnseenNet: LSDA based Detector with Online Training using only Image-Level labels: We proposed LSDA based detector, "UnseenNet", for the training of unseen classes using only image-level labels, i.e., training with no bounding box annotations. This model utilized the fastest classification and detection models (unlike LSDA) while using object detection and image classification datasets consisting of limited vocabulary.

Derivation of Minimum and Maximum limits of Response-Time for Weakly Supervised Learning: Besides devising a fast detector *UnseenNet*, we also derived the limits of *response-time* from 5-min to 20-min in the area of weakly supervised learning (i.e., training with no bounding boxes), where existing frameworks take >5.5-hours to attain similar mAP.

9.4 Limitations & Open Questions

The research conducted in this thesis opens multiple dimensions and thus also recognized the following emerging limitations with associated questions:

- **Subscriptions are only Keywords**: Presently, we allowed subscriptions that consist of only keywords, and we have not explored the expressive power of event processing languages. Thus our model would fail to support any query consisting of any complex operation within subscriptions.
- **Quality of Data**: Since we used standardized object detection and image classification datasets, our models' performance is not known for images captured under different conditions like weather conditions, light, low-resolution images, etc. What will be the impact of noise in terms of the accuracy and response time for such images is still an open question. Moreover, if we download image data from the Web, then in-spite of using certified GoogleImageDownloader, there are chances to get some unreliable data. Presently, we are checking whether the image is corrupt in our models before training, but how beneficial it would be the pre-checking the content of the image is not the scope of our work. And, consequently, what will be the time complexity of such security measures is essential to look at before incorporating them.
- **Tuning of model-specific hyperparameters**:In the hyperparameter-based model, we are changing only *optimization* hyperparameters (learning rate, batch size, and the number of epochs) recommended by experts to tune for effective training. However, we believe it would be worth analyzing *model-specific* hyperparameters fixed by neural-network-based models like architecture, image size, dropout, number of layers, etc.
- **Baseline Detectors are trained offline**: Strong and Weak Baseline detectors in "UnseenNet" are trained offline on 100 classes, and there is no provision of adding more classes in them in the future. What will be its effect on performance if we make their training dynamic (online) also. Moreover, how unseen classes'

performance would change with an increase in the number of classes in baseline detectors. Since baseline detector classes are chosen from available classes of object detection datasets which are very limited in number, there was no scope for considering the unique classes which are very different from each other. This limitation could be reduced by utilizing the detectors constructed by "UnseenNet" for different unseen classes.

- **Space Complexity**: We constructed classifiers one by one on request of unseen concepts. In our case, 1 million classifiers would take 1TB space. However, suppose the proposed model would be used with other computationally expensive object detection models; in that case, it may take more or maybe less space, depending on the object detection model's robustness. Other than analyzing the space complexity problem, it could be useful to analyze our model's performance with a multi-class classifier construction option. However, we assume multi-class classifiers would be beneficial only on receiving the request of multiple unseen concept-based subscriptions.

9.5 Future Research Directions

Some exciting areas that are derived from our work and useful for future research are as follows:

- **Evaluation on Videos/Music**: The multimedia event processing model that we proposed is generic and could work for videos or music-based multimedia data other than images by introducing a video/music-based multimedia based operators. It would be interesting to see the efficiency of proposed architectures other data and train classifier online for their unseen classes.
- **Inclusion of Multi-Class classifiers**: Investigation of training time of multi-class classifiers in contrast to binary classifiers is also a reasonable future direction to make the proposed approach more effective in terms of time and space.
- **Enhancement of Resources**: Presently, I used the most recent object detection models *SSD* [36], *YOLO* [35], and *RetinaNet* [37] for experiments. However, I assume it will be easy to incorporate new models with the proposed system in the future. This may require input, and output formats of object detection models should be the same. MMDetection [1] and Detectron [2] are open-source object detection toolboxes that could also be used for the interoperability of the proposed

[1] https://github.com/open-mmlab/mmdetection
[2] https://github.com/facebookresearch/Detectron

system. Similarly, for the online toolkits, I mainly used OpenImages [3] and Google Images Downloader [4]. However, other toolkits like ImageNet_Utils [5], Bing Scrapper [6], Flickr_Photos [7], etc., can also be used by calling their functions from the downloading module (specified in Chapter–8).

- **Effect of Size of Training/Testing Dataset**: In the present scenario, we used all available training images with bounding boxes of object detection datasets to train seen classes. However, current object detection datasets are biased towards few classes, and consist of a very uneven division of the number of images per class. The same is the case for testing the dataset. We believe it would be beneficial to use an equal number of images for each class while training and testing and derive the performance change.

- **Visual/Semantic Similarity**: As we stated in the "UnseenNet" model, we considered only the naive visual and semantic similarities to keep the response time low. We computed the Euclidean distance between the weights of the last layers of neural networks for the visual similarities. However, Tang et al. [30] proposed a new visual similarity measure that could also be incorporated into our UnseenNet in the future. Moreover, for the semantic similarity, we found the *path* vector of WordNet [29] more relevant in terms of unseen class differences, other than lch, wup, res, lin, jcn, etc. Since our work lacks the quantitative proof of *path* vector effectiveness, other operators or natural language processing-based methods also deserve investigation for effective multimedia event processing.

- **Scalability**: The ability of the event processing model to adapt with increasing load is referred to as scalability in literature. It is also described in event processing systems with the increase in the number of subscribers/subscriptions. Analyzing the scalability of the proposed multimedia event processing model and adding an adaptation module is a simple and worthwhile future direction for the deployment.

- **Devising of Neural-Network for Unseen Classes**: We incorporated the MobileNet within YOLOv3 for fast detection with fast classification. Adapting this network itself by changing (Adding/Deleting) layers may or may not reduced the training time on the cost of accuracy.

- **Infinite Vocabulary based Object Detection Dataset with no need of Bounding Boxes**: Our work also initiates an enhancement scheme for existing

[3] https://github.com/EscVM/OIDv4_ToolKit
[4] https://github.com/hardikvasa/google-images-download
[5] https://github.com/tzutalin/ImageNet_Utils
[6] https://github.com/funpokes/bing-image-search
[7] https://www.flickr.com/services/api/

object detection and image classification datasets having bounded (finite) vocabulary to turn it into the unbounded (infinite) vocabulary. With the use of our UnseenNet, millions of detectors can be trained for new classes, which then can create object detection datasets on testing by producing bounding boxes for images of new classes. Our contribution to this idea will open a new paradigm in the field of object detection datasets.

- **Response-Time based Strategies**: We have conducted experiments using minimum, maximum, and optimal response-time based strategies since those were highly distinguishable among themselves in analysing the best performance on the detection of multimedia events. We may also design more strategies in the future based on a higher rate of change, approximately-zero-response time, and constant-accuracy.

- **Approximation of Subscriptions/Events:** Optimization of subscriptions and event streams is another area for future research, which requires the knowledge modeling of IoMT generated data. We covered the optimization of subscriptions based on commonalities, which could be replaced with approximate event processing. Optimization of multimedia events could also be incorporated by dropping the repetitive frames in the multimedia streams. Such optimizations will contribute towards the effectiveness of our proposed approach in smart cities.

- **Unsupervised Learning for Unseen Classes**: Our weakly-supervised learning-based model "UnseenNet" can also be extended in the future for unsupervised learning to reduce the need for data for the large number of nearest-neighbor seen classes and computation of similarities for unseen classes.

Bibliography

[1] Cisco, Visual Networking Index: Forecast and Methodology, 2015–2020. The zettabyte era: Trends and analysis. *White Paper*, 2016. URL https://webobjects.cdw.com/webobjects/media/pdf/Solutions/Networking/White-Paper-Cisco-The-Zettabyte-Era-Trends-and-Analysis.pdf.

[2] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[3] Judy Hoffman, Sergio Guadarrama, Eric S Tzeng, Ronghang Hu, Jeff Donahue, Ross Girshick, Trevor Darrell, and Kate Saenko. Lsda: Large scale detection through adaptation. In *Advances in Neural Information Processing Systems*, pages 3536–3544, 2014.

[4] Kah Phooi Seng and Li-Minn Ang. A Big Data Layered Architecture and Functional Units for the Multimedia Internet of Things MIoT. *IEEE Transactions on Multi-Scale Computing Systems*, 2018.

[5] Sheeraz A Alvi, Bilal Afzal, Ghalib A Shah, Luigi Atzori, and Waqar Mahmood. Internet of multimedia things: Vision and challenges. *Ad Hoc Networks*, 33:87–111, 2015.

[6] Sufyan Almajali, I Dhiah el Diehn, Haythem Bany Salameh, Moussa Ayyash, and Hany Elgala. A distributed multi-layer mec-cloud architecture for processing large scale iot-based multimedia applications. *Multimedia Tools and Applications*, pages 1–22, 2018.

[7] Qin Wang, Yanxiao Zhao, Wei Wang, Daniel Minoli, Kazem Sohraby, Hongbo Zhu, and Ben Occhiogrosso. Multimedia iot systems and applications. In *Global Internet of Things Summit (GIoTS), 2017*, pages 1–6. IEEE, 2017.

[8] Gianpaolo Cugola and Alessandro Margara. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)*, 44 (3):15, 2012.

[9] Edward Curry, Schahram Dustdar, Quan Z Sheng, and Amit Sheth. Smart cities–enabling services and applications. *Journal of Internet Services and Applications*, 7(1):6, 2016.

[10] Holger Glasl, David Schreiber, Nikolaus Viertl, Stephan Veigl, and Gustavo Fernandez. Video based traffic congestion prediction on an embedded system. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 950–955. IEEE, 2008.

[11] Chiao-Fe Shu, Arun Hampapur, Max Lu, Lisa Brown, Jonathan Connell, Andrew Senior, and Yingli Tian. Ibm smart surveillance system (s3): a open and extensible framework for event based surveillance. In *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, pages 318–323. IEEE, 2005.

[12] Pirkko Mustamo. Object detection in sports: TensorFlow Object Detection API case study. *University of Oulu*, 2018. URL http://jultika.oulu.fi/files/nbnfioulu-201802081173.pdf.

[13] Xiu-Shen Wei, Bin-Bin Gao, and Jianxin Wu. Deep spatial pyramid ensemble for cultural event recognition. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 38–44, 2015.

[14] Andreas Kamilaris and Francesc X Prenafeta-Boldú. Disaster monitoring using unmanned aerial vehicles and deep learning. *arXiv preprint arXiv:1807.11805*, 2018.

[15] Mrigank Rochan and Yang Wang. Weakly supervised localization of novel objects using appearance transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4315–4324, 2015.

[16] Aluizio Rocha Neto, Thiago P Silva, Thais Batista, Flávia C Delicato, Paulo F Pires, and Frederico Lopes. Leveraging edge intelligence for video analytics in smart city applications. *Information*, 12(1):14, 2021.

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[19] Indre Zliobaite and Bogdan Gabrys. Adaptive preprocessing for streaming data. *IEEE transactions on knowledge and data Engineering*, 26(2):309–321, 2014.

[20] Haixun Wang, Wei Fan, Philip S Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 226–235. AcM, 2003.

[21] Dewan Md Farid, Li Zhang, Alamgir Hossain, Chowdhury Mofizur Rahman, Rebecca Strachan, Graham Sexton, and Keshav Dahal. An adaptive ensemble classifier for mining concept drifting data streams. *Expert Systems with Applications*, 40(15):5895–5906, 2013.

[22] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017.

[23] Chris Thornton, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 847–855. ACM, 2013.

[24] James Bergstra, Dan Yamins, and David D Cox. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in Science Conference*, pages 13–20. Citeseer, 2013.

[25] Jonas Vlasselaer, Wannes Meert, and Marian Verhelst. Towards resource-efficient classifiers for always-on monitoring. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 305–321. Springer, 2018.

[26] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[28] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github. com/openimages*, 2:3, 2017.

[29] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics, 2004.

[30] Yuxing Tang, Josiah Wang, Boyang Gao, Emmanuel Dellandréa, Robert Gaizauskas, and Liming Chen. Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2119–2128, 2016.

[31] Asra Aslam and Edward Curry. A survey on object detection for the internet of multimedia things (iomt) using deep learning and event-based middleware: Approaches, challenges, and future directions. *Image and Vision Computing*, 106: 104095, 2021.

[32] Asra Aslam and Edward Curry. Towards a generalized approach for deep neural network based event processing for the internet of multimedia things. *IEEE Access*, 6:25573–25587, 2018.

[33] Asra Aslam and Edward Curry. Investigating response time and accuracy in online classifier learning for multimedia publish-subscribe systems. *Multimedia Tools and Applications*, pages 1–37, 2021.

[34] Asra Aslam and Edward Curry. Reducing response time for multimedia event processing using domain adaptation. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pages 261–265, 2020.

[35] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[36] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[37] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[38] Asra Aslam. Object detection for unseen domains while reducing response time using knowledge transfer in multimedia event processing. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pages 373–377, 2020.

[39] Edward Curry and Amit Sheth. Next-generation smart environments: From system of systems to data ecosystems. *IEEE Intelligent Systems*, 33(3):69–76, 2018. doi: 10.1109/MIS.2018.033001418.

[40] Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards recognizing unseen categories in unseen domains. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 466–483. Springer, 2020.

[41] Michael Stonebraker, Uğur Çetintemel, and Stan Zdonik. The 8 requirements of real-time stream processing. *ACM Sigmod Record*, 34(4):42–47, 2005.

[42] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[43] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1446–1455, 2019.

[44] Nikhil Yadav and Utkarsh Binay. Comparative study of object detection algorithms. *International Research Journal of Engineering and Technology (IRJET)*, 4(11):586–591, 2017.

[45] Archit Gupta, Raghav Puri, Mrinal Verma, Siddharth Gunjyal, and Ashish Kumar. Performance comparison of object detection algorithms with different feature extractors. In *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 472–477. IEEE, 2019.

[46] Shivang Agarwal, Jean Ogier Du Terrail, and Frédéric Jurie. Recent advances in object detection in the age of deep convolutional neural networks. *arXiv preprint arXiv:1809.03193*, 2018.

[47] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 2019.

[48] Mohammad Abdur Razzaque, Marija Milojevic-Jevric, Andrei Palade, and Siobhán Clarke. Middleware for internet of things: a survey. *IEEE Internet of things journal*, 3(1):70–95, 2015.

[49] Edward Curry. Message-oriented middleware. *Middleware for communications*, pages 1–28, 2004.

[50] Kashif Ahmad and Nicola Conci. How deep features have improved event recognition in multimedia: a survey. *ACM Transactions on Multimedia Computing Communications and Applications*, January, 2019.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[52] Min Wu and Marta Kwiatkowska. Robustness guarantees for deep neural networks on videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 311–320, 2020.

[53] Sumi Helal, Flavia C Delicato, Cintia B Margi, Satyajayant Misra, and Markus Endler. Challenges and opportunities for data science and machine learning in iot systems-a timely debate: Part 1. *IEEE Internet of Things Magazine*, 2020.

[54] Hao Li, Hong Zhang, Xiaojuan Qi, Ruigang Yang, and Gao Huang. Improved techniques for training adaptive deep networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1891–1900, 2019.

[55] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020.

[56] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[57] Yunhui Guo, Yandong Li, Liqiang Wang, and Tajana Rosing. Depthwise convolution is all you need for learning multiple visual domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8368–8375, 2019.

[58] Kevin Bascol, Rémi Emonet, and Elisa Fromont. Improving domain adaptation by source selection. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3043–3047. IEEE, 2019.

[59] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009, 2018.

[60] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3339–3348, 2018.

[61] Yuxing Tang, Josiah Wang, Xiaofang Wang, Boyang Gao, Emmanuel Dellandréa, Robert Gaizauskas, and Liming Chen. Visual and semantic knowledge transfer

for large scale semi-supervised object detection. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3045–3058, 2017.

[62] Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G Macready. A robust learning approach to domain adaptive object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 480–490, 2019.

[63] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13846–13855, 2020.

[64] Yang Shu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Transferable curriculum for weakly-supervised domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4951–4958, 2019.

[65] Seiichi Kuroki, Nontawat Charoenphakdee, Han Bao, Junya Honda, Issei Sato, and Masashi Sugiyama. Unsupervised domain adaptation based on source-guided discrepancy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4122–4129, 2019.

[66] Judith Hoffman. *Adaptive learning algorithms for transferable visual recognition.* University of California, Berkeley, 2016.

[67] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition CVPR 2009.*, pages 248–255. IEEE, 2009.

[68] Ovidiu Vermesan, Peter Friess, et al. *Internet of things-from research and innovation to market deployment*, volume 29. River publishers Aalborg, 2014.

[69] European Commission and EPoSS. Internet of Things in 2020, Roadmap for the Future. Version 1.1, INFSO D.4 Networked Enterprise & RFID INFSO G.2 Micro & Nanosystems, in co-operation with the Working Group RFID of the ETP EPOSS, May 27 2008. URL http://www.iot-visitthefuture.eu/fileadmin/documents/researchforeurope/270808_IoT_in_2020_Workshop_Report_V1-1.pdf.

[70] Harald Sundmaeker, Patrick Guillemin, Peter Friess, and Sylvie Woelfflé. Vision and challenges for realising the internet of things. *Cluster of European Research Projects on the Internet of Things, European Commision*, 3(3):34–36, 2010.

[71] Ovidiu Vermesan, Peter Friess, Patrick Guillemin, Harald Sundmaeker, Markus Eisenhauer, Klaus Moessner, Franck Le Gall, and Philippe Cousin. Internet of things strategic research and innovation agenda. In *Internet of things: converging*

technologies for smart environments and integrated ecosystems, pages 7–152. River Publishers, 2013.

[72] Ken Sakamura. Challenges in the age of ubiquitous computing: a case study of t-engine, an open development platform for embedded systems. In *Proceedings of the 28th international conference on Software engineering*, pages 713–720. ACM, 2006.

[73] A. Gluhak M. Presser. The internet of things: Connecting the real world with the digital world. *EURESCOM mess@ge – The Magazine for Telecom Insiders*, 2, 2009. URL http://www.eurescom.eu/message.

[74] Maarten Botterman. Internet of things: an early reality of the future internet. In *Workshop Report, European Commission Information Society and Media*, volume 15, 2009.

[75] Bruce Sterling. Shaping things. mediawork pamphlet series, 2005.

[76] ITU Internet Reports. The internet of things. Technical report, November 2005.

[77] A. Dunkels and J.P. Vasseur. IP for smart objects, internet protocol for smart objects (ipso) alliance. *White Paper*, 1, September 2008. URL http://dunkels.com/adam/dunkels08ipso.pdf.

[78] Neil Gershenfeld, Raffi Krikorian, and Danny Cohen. The internet of things. *Scientific American*, 291(4):76–81, 2004.

[79] Dominique Guinard and Vlad Trifa. Towards the web of things: Web mashups for embedded devices. In *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain*, volume 15, 2009.

[80] Ioan Toma, Elena Simperl, and Graham Hench. A joint roadmap for semantic technologies and the internet of things. In *Proceedings of the Third STI Roadmapping Workshop, Crete, Greece*, volume 1, pages 140–53, 2009.

[81] Artem Katasonov, Olena Kaykova, Oleksiy Khriyenko, Sergiy Nikitin, and Vagan Y Terziyan. Smart semantic middleware for the internet of things. *Icinco-Icso*, 8:169–178, 2008.

[82] W Wahlster. Web 3.0: semantic technologies for the internet of services and of things. *Lecture at the Dresden Future Forum*, pages 100–106, June 2008.

[83] Iñaki Vázquez. Social devices: Semantic technology for the internet of things. *Week@ ESI, Zamudio, Spain*, 2009.

[84] Edward Curry. *Real-time linked dataspaces: Enabling data ecosystems for intelligent systems*. Springer Nature, 2020. doi: 10.1007/978-3-030-29665-0.

[85] Edward Curry, Wassim Derguech, Souleiman Hasan, Christos Kouroupetroglou, and Umair ul Hassan. A real-time linked dataspace for the internet of things: enabling "pay-as-you-go" data management in smart environments. *Future Generation Computer Systems*, 90:405–422, 2019.

[86] Alessandro Floris and Luigi Atzori. Quality of experience in the multimedia internet of things: Definition and practical use-cases. In *Communication Workshop (ICCW), 2015 IEEE International Conference on*, pages 1747–1752. IEEE, 2015.

[87] Titus Balan, Dan Robu, and Florin Sandu. Multihoming for mobile internet of multimedia things. *Mobile Information Systems*, 2017, 2017.

[88] Jinfei Yang, Jiajia Li, and Shouqiang Liu. A new algorithm of stock data mining in internet of multimedia things. *The Journal of Supercomputing*, pages 1–16, 2017.

[89] Hassan Noura, Ali Chehab, Lama Sleem, Mohamad Noura, Raphaël Couturier, and Mohammad M. Mansour. One round cipher algorithm for multimedia iot devices. *Multimedia tools and applications*, 2018. doi: 10.1007/s11042-018-5660-y.

[90] Maher Jridi, Thibault Chapel, Victor Dorez, Guénolé Le Bougeant, and Antoine Le Botlan. Soc-based edge computing gateway in the context of the internet of multimedia things: Experimental platform. *Journal of Low Power Electronics and Applications*, 8(1):1, 2018.

[91] Ali Nauman, Yazdan Ahmad Qadri, Muhammad Amjad, Yousaf Bin Zikria, Muhammad Khalil Afzal, and Sung Won Kim. Multimedia internet of things: A comprehensive survey. *IEEE Access*, 8:8202–8250, 2020.

[92] Liang Zhou and Han-Chieh Chao. Multimedia traffic security architecture for the internet of things. *IEEE Network*, 25(3), 2011.

[93] Malaram Kumhar, Gaurang Raval, and Vishal Parikh. Quality evaluation model for multimedia internet of things (miot) applications: Challenges and research directions. In *International Conference on Internet of Things and Connected Technologies*, pages 330–336. Springer, 2019.

[94] Shancang Li, Li Da Xu, and Shanshan Zhao. The internet of things: a survey. *Information Systems Frontiers*, 17(2):243–259, 2015.

[95] Felix Wortmann and Kristina Flüchter. Internet of things. *Business & Information Systems Engineering*, 57(3):221–224, 2015.

[96] Javier Silvestre-Blanes, Víctor Sempere-Payá, and Teresa Albero-Albero. Smart sensor architectures for multimedia sensing in iomt. *Sensors*, 20(5):1400, 2020.

[97] Raphael Troncy, Benoit Huet, and Simon Schenk. *Multimedia semantics: Metadata, analysis and interaction*. John Wiley & Sons, 2011.

[98] Hari Sundaram, Lexing Xie, Munmun De Choudhury, Yu-Ru Lin, and Apostol Natsev. Multimedia semantics: Interactions between content and community. *Proceedings of the IEEE*, 100(9):2737–2758, 2012.

[99] F-M Nack, Hazel Lynda Hardman, Frank Nack, and Lynda Hardman. Towards a syntax for multimedia semantics. Technical report, Centrumvoor Wiskunde en Informatica (CWI), Amsterdam, 2002.

[100] Gary Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[101] Gaurav Aggarwal, Pradeep Kumar Dubey, Sugata Ghosal, Ashutosh Kulshreshtha, and Tumkur Venkatanarayana Rao Ravi. Interactive framework for understanding user's perception of multimedia data, June 18 2002. US Patent 6,408,293.

[102] Zhigang Ma, Feiping Nie, Yi Yang, Jasper RR Uijlings, Nicu Sebe, and Alexander G Hauptmann. Discriminating joint feature analysis for multimedia data understanding. *IEEE Transactions on Multimedia*, 14(6):1662–1672, 2012.

[103] Yi Wu, Edward Y Chang, Kevin Chen-Chuan Chang, and John R Smith. Optimal multimodal fusion for multimedia data analysis. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 572–579. ACM, 2004.

[104] Mark T Maybury. *Intelligent multimedia information retrieval*. Association for the Advancement of Artificial Intelligence (AAAI) Press, 1997.

[105] Yonghong Tian, Jaideep Srivastava, Tiejun Huang, and Noshir Contractor. Social multimedia computing. *Computer*, 43(8):27–36, 2010.

[106] Shu-Ching Chen. Multimedia databases and data management: a survey. *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, 1(1):1–11, 2010.

[107] TN Manjunath, Ravindra S Hegadi, and GK Ravikumar. A survey on multimedia data mining and its relevance today. *International Journal of Computer Science and Network Security (IJCSNS)*, 10(11):165–170, 2010.

[108] Ling Guan, Yifeng He, and Sun-Yuan Kung. *Multimedia image and video processing*. CRC press, 2012.

[109] Peter Pirsch and H-J Stolberg. Vlsi implementations of image and video multimedia processing systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(7):878–891, 1998.

[110] Mihaela Van der Schaar and Philip A Chou. *Multimedia over IP and wireless networks: compression, networking, and systems*. Elsevier, 2011.

[111] Kamisetty Ramamohan Rao, Zoran S Bojkovic, and Dragorad A Milovanovic. *Multimedia communication systems: techniques, standards, and networks*. Prentice Hall PTR, 2002.

[112] Gerard Richter and Jean-Yves Solves. Multimedia processing system architecture, April 20 2004. US Patent 6,725,279.

[113] Richard V Cox, Barry G Haskell, Yann LeCun, Behzad Shahraray, and Lawrence Rabiner. On the applications of multimedia processing to communications. *Proceedings of the IEEE*, 86(5):755–824, 1998.

[114] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[115] Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864, 2010.

[116] Jyrki Kivinen, Alexander J Smola, and Robert C Williamson. Online learning with kernels. *IEEE transactions on signal processing*, 52(8):2165–2176, 2004.

[117] Steven CH Hoi, Jialei Wang, and Peilin Zhao. Libol: A library for online learning algorithms. *The Journal of Machine Learning Research*, 15(1):495–499, 2014.

[118] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. Identifying suspicious urls: an application of large-scale online learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 681–688. ACM, 2009.

[119] Alem Čolaković and Mesud Hadžialić. Internet of things (iot): A review of enabling technologies, challenges, and open research issues. *Computer Networks*, 144:17–39, 2018.

[120] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[121] Yushu Zhang, Qi He, Yong Xiang, Leo Yu Zhang, Bo Liu, Junxin Chen, and Yiyuan Xie. Low-cost and confidentiality-preserving data acquisition for internet of multimedia things. *IEEE Internet of Things Journal*, 2017.

[122] Aparna Kumari, Sudeep Tanwar, Sudhanshu Tyagi, Neeraj Kumar, Michele Maasberg, and Kim-Kwang Raymond Choo. Multimedia big data computing and internet of things applications: A taxonomy and process model. *Journal of Network and Computer Applications*, 124:169–195, 2018.

[123] Anselmo Luiz Éden Battisti, Débora Christina Muchaluat-Saade, and Flávia C Delicato. Enabling internet of media things with edge-based virtual multimedia sensors. *IEEE Access*, 2021.

[124] Ian F Akyildiz, Tommaso Melodia, and Kaushik R Chowdhury. A survey on wireless multimedia sensor networks. *Computer networks*, 51(4):921–960, 2007.

[125] Anselmo Luiz Éden Battisti, Débora Christina Muchaluat-Saade, and Flávia C Delicato. V-prism: An edge-based iot architecture to virtualize multimedia sensors. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, pages 1–6. IEEE, 2020.

[126] Ahmed Elshafeey, Neamat S Abd Elkader, and M Zorkany. Compressed sensing video streaming for internet of multimedia things. *International Journal of Cyber-Security and Digital Forensics*, 6(1):44–54, 2017.

[127] Alessandro Floris and Luigi Atzori. Managing the quality of experience in the multimedia internet of things: A layered-based approach. *Sensors*, 16(12):2057, 2016.

[128] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[129] Stephan Kudyba. *Big data, mining, and analytics: components of strategic decision making*. CRC Press, 2014.

[130] Seref Sagiroglu and Duygu Sinanc. Big data: A review. In *2013 International Conference on Collaboration Technologies and Systems (CTS)*, pages 42–47. IEEE, 2013.

[131] Chun-Wei Tsai, Chin-Feng Lai, Han-Chieh Chao, and Athanasios V Vasilakos. Big data analytics: a survey. *Journal of Big data*, 2(1):21, 2015.

[132] Javier Molina, Javier M Mora-Merchan, Julio Barbancho, and Carlos Leon. Multimedia data processing and delivery in wireless sensor networks. In *Wireless Sensor Networks: Application-Centric Design*. InTech, 2010.

[133] Ian F Akyildiz, Tommaso Melodia, and Kaushik R Chowdhury. Wireless multimedia sensor networks: Applications and testbeds. *Proceedings of the IEEE*, 96 (10):1588–1605, 2008.

[134] Jiachen Yang, Shudong He, Yancong Lin, and Zhihan Lv. Multimedia cloud transmission and storage system based on internet of things. *Multimedia Tools and Applications*, 76(17):17735–17750, 2017.

[135] Souleiman Hasan. *Loose coupling in heterogeneous event-based systems via approximate semantic matching and dynamic enrichment*. PhD thesis, 2016.

[136] Kiev Gama, Lionel Touseau, and Didier Donsez. Combining heterogeneous service technologies for building an internet of things middleware. *Computer Communications*, 35(4):405–417, 2012.

[137] Ahmed Aliyu, Abdul H Abdullah, Omprakash Kaiwartya, Yue Cao, Jaime Lloret, Nauman Aslam, and Usman Mohammed Joda. Towards video streaming in iot environments: Vehicular communication perspective. *Computer Communications*, 2017.

[138] Thiago Teixeira, Dimitrios Lymberopoulos, Eugenio Culurciello, Yiannis Aloimonos, and Andreas Savvides. A lightweight camera sensor network operating on symbolic information. In *Proceedings of the 1st Workshop on Distributed Smart Cameras*, 2006.

[139] Stephan Hengstler and Hamid Aghajan. Wisnap: a wireless image sensor network application platform. In *2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006.*, pages 6–pp. IEEE, 2006.

[140] Zhiyuan Wang, Jifeng Huang, Neal N Xiong, Xiaoping Zhou, Xiao Lin, and Theodore Lee Ward. A robust vehicle detection scheme for intelligent traffic surveillance systems in smart cities. *IEEE Access*, 2020.

[141] Susanne Boll, Jochen Meyer, and Noel E O'Connor. Health media: From multimedia signals to personal health insights. *IEEE MultiMedia*, 25(1):51–60, 2018.

[142] Cihan Küçükkeçeci et al. Big data model simulation on a graph database for surveillance in wireless multimedia sensor networks. *Big Data Research*, 2017.

[143] Jun Huang, Qiang Duan, Yanxiao Zhao, Zhong Zheng, and Wei Wang. Multicast routing for multimedia communications in the internet of things. *IEEE Internet of Things Journal*, 4(1):215–224, 2017.

[144] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3586–3595, 2020.

[145] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015.

[146] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[147] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[148] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[149] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[150] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[151] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[152] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[153] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[154] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, pages 1–26, 2020.

[155] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[156] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE international conference on computer vision*, pages 4918–4927, 2019.

[157] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[158] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 49–58, 2016.

[159] Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*, 2014.

[160] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[161] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065*, 2016.

[162] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[163] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29:3630–3638, 2016.

[164] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

[165] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019.

[166] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[167] Kun Fu, Tengfei Zhang, Yue Zhang, Menglong Yan, Zhonghan Chang, Zhengyuan Zhang, and Xian Sun. Meta-ssd: Towards fast adaptation for few-shot object detection with meta-learning. *IEEE Access*, 7:77597–77606, 2019.

[168] Tao Wang, Xiaopeng Zhang, Li Yuan, and Jiashi Feng. Few-shot adaptive faster r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7173–7182, 2019.

[169] Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. Few-example object detection with model communication. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1641–1654, 2018.

[170] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8420–8429, 2019.

[171] Yuhao Zhang and Arun Kumar. Panorama: a data system for unbounded vocabulary querying over video. *Proceedings of the VLDB Endowment*, 13(4):477–491, 2019.

[172] Autonomic Computing et al. An architectural blueprint for autonomic computing. *IBM White Paper*, 31:1–6, 2006.

[173] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[174] Patrick Th Eugster, Pascal A Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys (CSUR)*, 35 (2):114–131, 2003.

[175] René Meier and Vinny Cahill. Taxonomy of distributed event-based programming systems. *The Computer Journal*, 48(5):602–626, 2005.

[176] Chinnapong Angsuchotmetee and Richard Chbeir. A survey on complex event definition languages in multimedia sensor networks. In *Proceedings of the 8th International Conference on Management of Digital EcoSystems*, pages 99–108. ACM, 2016.

[177] Antonio Carzaniga, David S Rosenblum, and Alexander L Wolf. Achieving scalability and expressiveness in an internet-scale event notification service. In *Proceedings of ACM symposium on Principles of distributed computing*, pages 219–227. ACM, 2000.

[178] Jean Bacon, Ken Moody, John Bates, Chaoying Ma, A McNeil, O Seidel, and M Spiteri. Generic support for distributed applications. *Computer*, 33(3):68–76, 2000.

[179] Antonio Carzaniga, David S Rosenblum, and Alexander L Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems (TOCS)*, 19(3):332–383, 2001.

[180] Mads Haahr, René Meier, Paddy Nixon, Vinny Cahill, and Eric Jul. Filtering and scalability in the ECO distributed event model. In *Software Engineering for Parallel and Distributed Systems, International Symposium on*, pages 83–83. IEEE Computer Society, 2000.

[181] Gradimir Starovic, Vinny Cahill, and Brendan Tangney. An event based object model for distributed programming. In *International Conference on Object Oriented Information Systems (OOIS'95)*, pages 72–86. Springer, 1996.

[182] Gianpaolo Cugola, Elisabetta Di Nitto, and Alfonso Fuggetta. The jedi event-based infrastructure and its application to the development of the opss wfms. *IEEE transactions on Software Engineering*, 27(9):827–850, 2001.

[183] Gianpaolo Cugola, H Jacobsen, et al. Using publish/subscribe middleware for mobile systems. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(4):25–33, 2002.

[184] Ioana Burcea, H-A Jacobsen, Eyal De Lara, Vinod Muthusamy, and Milenko Petrovic. Disconnected operation in publish/subscribe middleware. In *IEEE International Conference on Mobile Data Management, 2004. Proceedings. 2004*, pages 39–50. IEEE, 2004.

[185] Milenko Petrovic, Ioana Burcea, and Hans-Arno Jacobsen. S-topss: Semantic toronto publish/subscribe system. In *VLDB '03: Proceedings of the 29th international conference on Very large data bases*, pages 1101–1104. Elsevier, 2003.

[186] Peter R Pietzuch and Jean M Bacon. Hermes: A distributed event-based middleware architecture. In *Proceedings 22nd International Conference on Distributed Computing Systems Workshops*, pages 611–618. IEEE, 2002.

[187] René Meier and Vinny Cahill. Steam: Event-based middleware for wireless ad hoc networks. In *Proceedings 22nd International Conference on Distributed Computing Systems Workshops*, pages 639–644. IEEE, 2002.

[188] René Meier and Vinny Cahill. Location-aware event-based middleware: A paradigm for collaborative mobile applications? In *8th CaberNet Radicals Workshop*, Ajaccio, Corsica, France, October 2003.

[189] René Meier, Barbara Hughes, Raymond Cunningham, and Vinny Cahill. Towards real-time middleware for applications of vehicular ad hoc networks. In *IFIP International Conference on Distributed Applications and Interoperable Systems*, pages 1–13. Springer, 2005.

[190] René Meier, Vinny Cahill, Andronikos Nedos, and Siobhán Clarke. Proximity-based service discovery in mobile ad hoc networks. In *IFIP International Conference on Distributed Applications and Interoperable Systems*, pages 115–129. Springer, 2005.

[191] Jinling Wang, Beihong Jin, and Jing Li. An ontology-based publish/subscribe system. In *Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, pages 232–253. Springer-Verlag, 2004.

[192] Haifeng Liu and H-Arno Jacobsen. A-topss—a publish/subscribe system supporting approximate matching. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, pages 1107–1110. Elsevier, 2002.

[193] Haifeng Liu and Hans-Arno Jacobsen. A-topss: a publish/subscribe system supporting imperfect information processing. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 1281–1284. VLDB Endowment, 2004.

[194] Haifeng Liu and H-A Jacobsen. Modeling uncertainties in publish/subscribe systems. In *Proceedings. 20th International Conference on Data Engineering*, pages 510–521. IEEE, 2004.

[195] Thirunavukkarasu Sivaharan, Gordon Blair, and Geoff Coulson. Green: A configurable and re-configurable publish-subscribe middleware for pervasive computing. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 732–749. Springer, 2005.

[196] Mirco Musolesi, Cecilia Mascolo, and Stephen Hailes. Emma: Epidemic messaging middleware for ad hoc networks. *Personal and Ubiquitous Computing*, 10(1):28–36, 2005.

[197] Eduardo Souto, Germano Guimarães, Glauco Vasconcelos, Mardoqueu Vieira, Nelson Rosa, Carlos Ferraz, and Judith Kelner. Mires: a publish/subscribe middleware for sensor networks. *Personal and Ubiquitous Computing*, 10(1):37–44, 2005.

[198] Admilson RL Ribeiro, Fabio Silva, Lilian C Freitas, João Crisóstomo Costa, and Carlos R Francês. Sensorbus: a middleware model for wireless sensor networks. In *Proceedings of the 3rd international IFIP/ACM Latin American conference on Networking*, pages 1–9. ACM, 2005.

[199] Kavi K Khedo and RK Subramanian. Meeca: Misense energy efficient clustering algorithm. In *2007 Third International Conference on Wireless Communication and Sensor Networks*, pages 31–35. IEEE, 2007.

[200] KK Khedo and R Subramanian. Misense: A generic energy efficient middleware architecture for wireless sensor networks. In *Proc. 2nd Int. Conf. Wireless Commun. Sensor Netw.(WCSN'06)*, pages 207–215, 2006.

[201] Ludger Fiege, Mira Mezini, Gero Mühl, and Alejandro P Buchmann. Engineering event-based systems with scopes. In *European Conference on Object-Oriented Programming*, pages 309–333. Springer, 2002.

[202] Ludger Fiege, Felix C Gartner, Oliver Kasten, and Andreas Zeidler. Supporting mobility in content-based publish/subscribe middleware. In *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 103–122. Springer, 2003.

[203] Ludger Fiege, Mariano Cilia, Gero Muhl, and Alejandro Buchmann. 'publish-subscribe grows up: support for management, visibility control, and heterogeneity. *IEEE Internet Computing*, 10(1):48–55, 2006.

[204] Ludger Fiege and G Mühl. Rebeca event-based electronic commerce architecture. *h ttp://www. gkec. informatik. tu-darmstadt. de/rebeca S*, 2000.

[205] Steven Lai, Jiannong Cao, and Yuan Zheng. Psware: A publish/subscribe middleware supporting composite event in wireless sensor network. In *2009 IEEE International Conference on Pervasive Computing and Communications*, pages 1–6. IEEE, 2009.

[206] Pruet Boonma and Junichi Suzuki. Tinydds: An interoperable and configurable publish/subscribe middleware for wireless sensor networks. In *Wireless Technologies: Concepts, Methodologies, Tools and Applications*, pages 819–846. IGI Global, 2012.

[207] José R Silva, Flávia C Delicato, Luci Pirmez, Paulo F Pires, Jesus MT Portocarrero, Taniro C Rodrigues, and Thais V Batista. Prisma: A publish-subscribe and resource-oriented middleware for wireless sensor networks. In *Proceedings of the Tenth Advanced International Conference on Telecommunications, Paris, France*, volume 2024, page 8797. Citeseer, 2014.

[208] Roy T Fielding and Richard N Taylor. *Architectural styles and the design of network-based software architectures*, volume 7. University of California, Irvine Doctoral dissertation, 2000.

[209] Souleiman Hasan, Sean O'Riain, and Edward Curry. Approximate semantic matching of heterogeneous events. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, pages 252–263. ACM, 2012.

[210] Souleiman Hasan and Edward Curry. Thingsonomy: Tackling variety in internet of things events. *IEEE Internet Computing*, 19(2):10–18, 2015.

[211] Roberto Baldoni and Antonino Virgillito. Distributed event routing in publish/-subscribe communication systems: a survey. *DIS, Universita di Roma La Sapienza, Tech. Rep*, 5, 2005.

[212] Satyen Kale, Elad Hazan, Fengyun Cao, and Jaswinder Pal Singh. Analysis and algorithms for content-based event matching. In *Distributed Computing Systems Workshops, 2005. 25th IEEE International Conference on*, pages 363–369. IEEE, 2005.

[213] Walid Rjaibi, Klaus R Dittrich, and Dieter Jaepel. Event matching in symmetric subscription systems. In *Proceedings of the 2002 conference of the Centre for Advanced Studies on Collaborative research*, page 9. IBM Press, 2002.

[214] Françoise Fabret, H Arno Jacobsen, François Llirbat, João Pereira, Kenneth A Ross, and Dennis Shasha. Filtering algorithms and implementation for very fast publish/subscribe systems. In *ACM Sigmod Record*, volume 30, pages 115–126. ACM, 2001.

[215] Tak W Yan and Héctor García-Molina. Index structures for selective dissemination of information under the boolean model. *ACM Transactions on Database Systems (TODS)*, 19(2):332–364, 1994.

[216] João Pereira, Françoise Fabret, François Llirbat, and Dennis Shasha. Efficient matching for web-based publish/subscribe systems. In *International Conference on Cooperative Information Systems*, pages 162–173. Springer, 2000.

[217] Antonio Carzaniga and Alexander L Wolf. Forwarding in a content-based network. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 163–174. ACM, 2003.

[218] Alexis Campailla, Sagar Chaki, Edmund Clarke, Somesh Jha, and Helmut Veith. Efficient filtering in publish-subscribe systems using binary decision diagrams. In *Proceedings of the 23rd International Conference on Software Engineering*, pages 443–452. IEEE Computer Society, 2001.

[219] Marcos K Aguilera, Robert E Strom, Daniel C Sturman, Mark Astley, and Tushar D Chandra. Matching events in a content-based subscription system. In *Proceedings of the eighteenth annual ACM symposium on Principles of distributed computing*, pages 53–61. ACM, 1999.

[220] John Gough and Glenn Smith. Efficient recognition of events in a distributed system. *Australian computer science communications*, 17:173–179, 1995.

[221] Françoise Fabret, François Llirbat, Joao Pereira, and Dennis Shasha. Efficient matching for content-based publish/subscribe systems. In *Proc. of Int'l Conf. on Cooperative Information Systems (CoopIS)*, 2000.

[222] Ching-Hao Lai and Chia-Chen Yu. An efficient real-time traffic sign recognition system for intelligent vehicles with smart phones. In *Technologies and Applications of Artificial Intelligence (TAAI), 2010 International Conference on*, pages 195–202. IEEE, 2010.

[223] Anurag Kanungo, Ayush Sharma, and Chetan Singla. Smart traffic lights switching and traffic density calculation using video processing. In *Engineering and computational sciences (RAECS), 2014 recent advances in*, pages 1–6. IEEE, 2014.

[224] Tej Tharang Dandala, Vallidevi Krishnamurthy, and Rajan Alwan. Internet of vehicles (iov) for traffic management. In *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, pages 1–4. IEEE, 2017.

[225] Yan Ke, Rahul Sukthankar, and Martial Hebert. Efficient visual event detection using volumetric features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 166–173. IEEE, 2005.

[226] Gérard Medioni, Isaac Cohen, François Brémond, Somboon Hongeng, and Ramakant Nevatia. Event detection and analysis from video streams. *IEEE Transactions on pattern analysis and machine intelligence*, 23(8):873–889, 2001.

[227] Meonghun Lee, Jeonghwan Hwang, and Hyun Yoe. Agricultural production system based on iot. In *2013 IEEE 16Th international conference on computational science and engineering*, pages 833–837. IEEE, 2013.

[228] Noboru Babaguchi, Yoshihiko Kawai, and Tadahiro Kitahashi. Event based indexing of broadcasted sports video by intermodal collaboration. *IEEE transactions on Multimedia*, 4(1):68–75, 2002.

[229] Laura Lopez-Fuentes, Joost van de Weijer, Marc Bolanos, and Harald Skinnemoen. Multi-modal deep learning approach for flood detection. In *MediaEval*, 2017.

[230] Sheharyar Ahmad, Kashif Ahmad, Nasir Ahmad, and Nicola Conci. Convolutional neural networks for disaster images retrieval. In *MediaEval*, 2017.

[231] Ying Liu and Linzhi Wu. Geological disaster recognition on optical remote sensing images using deep learning. *Procedia Computer Science*, 91:566–575, 2016.

[232] Benjamin Bischke, Patrick Helber, Christian Schulze, Venkat Srinivasan, Andreas Dengel, and Damian Borth. The multimedia satellite task at mediaeval 2017. In *MediaEval*, 2017.

[233] Yong Tang, Congzhe Zhang, Renshu Gu, Peng Li, and Bin Yang. Vehicle detection and recognition for intelligent traffic surveillance system. *Multimedia tools and applications*, 76(4):5817–5832, 2017.

[234] Mikolaj E Kundegorski, Samet Akçay, Michael Devereux, Andre Mouton, and Toby P Breckon. On using feature descriptors as visual words for object detection within x-ray baggage security screening. 2016.

[235] Juan Carlos San Miguel and José M Martínez. Robust unattended and stolen object detection by fusing simple algorithms. In *2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, pages 18–25. IEEE, 2008.

[236] Jermsak Jermsurawong, Mian Umair Ahsan, Abdulhamid Haidar, Haiwei Dong, and Nikolaos Mavridis. Car parking vacancy detection and its application in 24-hour statistical analysis. In *2012 10th International Conference on Frontiers of Information Technology*, pages 84–90. IEEE, 2012.

[237] Piyush Yadav and Edward Curry. Vidcep: Complex event processing framework to detect spatiotemporal patterns in video streams. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2513–2522. IEEE, 2019.

[238] Tony CT Kuo and Arbee LP Chen. Content-based query processing for video databases. *IEEE Transactions on Multimedia*, 2(1):1–13, 2000.

[239] Chenglang Lu, Mingyong Liu, and Zongda Wu. Svql: A sql extended query language for video databases. *International Journal of Database Theory and Application*, 8(3):235–248, 2015.

[240] Mario Döller, Ruben Tous, Matthias Gruhne, Kyoungro Yoon, Masanori Sano, and Ian S Burnett. The mpeg query format: Unifying access to multimedia retrieval systems. *IEEE MultiMedia*, (4):82–95, 2008.

[241] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.

[242] Shai Shalev-Shwartz and Yoram Singer. *Online learning: Theory, algorithms, and applications*. PhD thesis, Hebrew University, 2007.

[243] Ekaba Bisong. Batch vs. online learning. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pages 199–201. Springer, 2019.

[244] Gang Luo. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 5(1):18, 2016.

[245] Chris Thornton, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Auto-weka: Automated selection and hyper-parameter optimization of classification algorithms. *CoRR, abs/1208.3719*, 2012.

[246] Brent Komer, James Bergstra, and Chris Eliasmith. Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn. In *ICML workshop on AutoML*, volume 9, page 50. Citeseer, 2014.

[247] Yue Wu, Steven CH Hoi, Chenghao Liu, Jing Lu, Doyen Sahoo, and Nenghai Yu. Sol: A library for scalable online learning algorithms. *Neurocomputing*, 260:9–12, 2017.

[248] Burr Settles. Active learning literature survey. Technical report, University of California, Santa Cruz., 2009.

[249] Brendan Collins, Jia Deng, Kai Li, and Li Fei-Fei. Towards scalable dataset construction: An active learning approach. In *European conference on computer vision*, pages 86–98. Springer, 2008.

[250] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *The International Machine Learning Society (ICML), Williamstown*, pages 441–448, 2001.

[251] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of WISCONSIN–Madison, 2005.

[252] Bozidara Cvetkovic, B Kaluza, M Luštrek, and Matjaz Gams. Semi-supervised learning for adaptation of human activity recognition classifier to the user. In *Proc. of Int. Joint Conf. on Artificial Intelligence (IJCAI), Barcelona, Catalonia, Spain*, pages 24–29. Citeseer, 2011.

[253] Oge Marques and Nitish Barman. Semi-automatic semantic annotation of images using machine learning techniques. In *International Semantic Web Conference*, pages 550–565. Springer, 2003.

[254] Alexei Zhukov, N Tomin, V Kurbatsky, Denis Sidorov, Daniil Panasetsky, and Aoife Foley. Ensemble methods of classification for power systems security assessment. *Applied Computing and Informatics*, 2017.

[255] Božidara Cvetković, Boštjan Kaluža, Matjaž Gams, and Mitja Luštrek. Adapting activity recognition to a person with multi-classifier adaptive training. *Journal of Ambient Intelligence and Smart Environments*, 7(2):171–185, 2015.

[256] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[257] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[258] J. Bergstra, D. Yamins, and D. D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages I–115–I–123. JMLR.org, 2013. URL http://dl.acm.org/citation.cfm?id=3042817.3042832.

[259] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

[260] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992.

[261] Prem Melville. *Creating diverse ensemble classifiers*. Computer Science Department, University of Texas at Austin, 2003.

[262] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[263] Albert HR Ko, Robert Sabourin, and Alceu Souza Britto Jr. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, 41(5):1718–1731, 2008.

[264] Doyen Sahoo, Quang Pham, Jing Lu, and Steven CH Hoi. Online deep learning: Learning deep neural networks on the fly. *arXiv preprint arXiv:1711.03705*, 2017.

[265] Xingquan Zhu, Wei Ding, S Yu Philip, and Chengqi Zhang. One-class learning and concept summarization for data streams. *Knowledge and Information Systems*, 28 (3):523–553, 2011.

[266] Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. Tunability: Importance of hyperparameters of machine learning algorithms. *J. Mach. Learn. Res.*, 20(53): 1–32, 2019.

[267] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.

[268] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[269] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[270] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *IEEE Access*, 7: 128837–128868, 2019.

[271] Foster Provost and R Kohavi. Glossary of terms. *Journal of Machine Learning*, 30(2-3):271–274, 1998.

[272] Kendrick Boyd, Kevin H Eng, and C David Page. Area under the precision-recall curve: point estimates and confidence intervals. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 451–466. Springer, 2013.

[273] Lilian Weng. Object detection part 4: Fast detection models. *lilianweng.github.io/lil-log*, 2018. URL http://lilianweng.github.io/lil-log/2018/12/27/object-detection-part-4.html.

[274] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pages 242–264. IGI Global, 2010.

[275] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[276] Gabriela Csurka. *A Comprehensive Survey on Domain Adaptation for Visual Applications*, pages 1–35. Springer International Publishing, Cham, 2017. ISBN 978-3-319-58347-1. doi: 10.1007/978-3-319-58347-1_1. URL https://doi.org/10.1007/978-3-319-58347-1_1.

[277] Oscar Beijbom. Domain adaptations for computer vision applications. *arXiv preprint arXiv:1211.4860*, 2012.

[278] Sinno Jialin Pan, Qiang Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[279] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[280] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[281] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 17–36, 2012.

[282] Limin Wang, Zhe Wang, Yu Qiao, and Luc Van Gool. Transferring deep object and scene representations for event recognition in still images. *International Journal of Computer Vision*, 126(2-4):390–409, 2018.

[283] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[284] Joseph Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013–2016.

[285] Brian Chu, Vashisht Madhavan, Oscar Beijbom, Judy Hoffman, and Trevor Darrell. Best practices for fine-tuning visual classifiers to new domains. In *European conference on computer vision*, pages 435–442. Springer, 2016.

[286] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Freezeout: Accelerate training by progressively freezing layers. *arXiv preprint arXiv:1706.04983*, 2017.

[287] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[288] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.

[289] Jasper Uijlings, Stefan Popov, and Vittorio Ferrari. Revisiting knowledge transfer for training object class detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1101–1110, 2018.

[290] Yan Li, Junge Zhang, Kaiqi Huang, and Jianguo Zhang. Mixed supervised object detection with robust objectness transfer. *IEEE transactions on pattern analysis and machine intelligence*, 41(3):639–653, 2018.

[291] Yi Zhu, Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Soft proposal networks for weakly supervised object localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1841–1850, 2017.

[292] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 685–694, 2015.

[293] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. Weakly supervised object detection with convex clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1081–1089, 2015.

[294] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016.

[295] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. Weakly supervised object detection with posterior regularization. *Proceedings of the British Machine Vision Conference BMVC 2014*, pages 1–12, 2014.

[296] Alexander Kolesnikov and Christoph H Lampert. Improving weakly-supervised object localization by micro-annotation. *arXiv preprint arXiv:1605.05538*, 2016.

[297] Chong Wang, Kaiqi Huang, Weiqiang Ren, Junge Zhang, and Steve Maybank. Large-scale weakly supervised object localization via latent category learning. *IEEE Transactions on Image Processing*, 24(4):1371–1385, 2015.

[298] Zhiyuan Shi, Parthipan Siva, and Tao Xiang. Transfer learning by ranking for weakly supervised object annotation. *arXiv preprint arXiv:1705.00873*, 2017.

[299] Zhaoyang Zeng, Bei Liu, Jianlong Fu, Hongyang Chao, and Lei Zhang. Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8292–8300, 2019.

[300] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):189–203, 2016.

[301] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Multi-fold mil training for weakly supervised object localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2409–2416, 2014.

[302] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2843–2851, 2017.

[303] Peng Tang, Xinggang Wang, Angtian Wang, Yongluan Yan, Wenyu Liu, Junzhou Huang, and Alan Yuille. Weakly supervised region proposal network and object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 352–368, 2018.

[304] Judy Hoffman, Deepak Pathak, Eric Tzeng, Jonathan Long, Sergio Guadarrama, Trevor Darrell, and Kate Saenko. Large scale visual recognition through adaptation using joint representation and multiple instance learning. *The Journal of Machine Learning Research*, 17(1):4954–4984, 2016.

[305] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[306] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al.

Searching for mobilenetv3. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1314–1324, 2019.

[307] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 580–587, 2014.