LEARNING FROM INCOMPLETE AND HETEROGENEOUS DATA

by

Poojankumar B. Oza

A dissertation submitted to Johns Hopkins University
in conformity with the requirements for the degree of
Doctor of Philosophy

Baltimore, Maryland
October, 2021

Abstract

Deep convolutional neural networks (DCNNs) have shown impressive performance improvements for object detection and recognition problems. However, a vast majority of DCNN-based recognition methods are designed with two key assumptions in mind, i.e., 1) the assumption that all categories are known a priori and 2) both training and test data are drawn from a similar distribution. However, in many real-world applications, these assumptions do not necessarily hold and limit the generalization capability of a recognition model.

Generally, incomplete knowledge of the world is present at training time, and unknown classes can be submitted to an algorithm during testing. If the visual system is trained assuming that all categories are known a priori, it would fail to identify these cases with unknown classes during testing. Ideally, the goal of a visual recognition system would be to reject samples from unknown classes and classify samples from known classes. In this thesis, we consider this constraint and evaluate visual recognition systems under two problem settings, i.e., one-class and multi-class novelty detection. In the one-class setting, the goal is to learn a visual recognition system from a single category and reject any other category samples as unknown during testing. Whereas, in multi-class classification the visual recognition system aims to

learn from multiple-categories and reject any other category sample that is not part of the training category set as unknown. With experiments on multiple benchmark datasets we show that the proposed recognition systems are able to perform better compared to existing approaches.

Furthermore, we also recognize that in many real world conditions training and testing data distributions are often different. Due to this, the performance of a visual recognition system drops significantly. This is commonly referred to as dataset bias or domain-shift which can be addressed using domain adaptation. In particular, we address unsupervised domain adaptation in which the idea is to utilize an additional set of unlabeled data sampled from a particular domain to help improve the performance in that respective domain. Various experiments on multiple domain adaptation benchmarks show that the proposed strategy is able to generalize better compared to existing methods in the literature.

Primary Reader and Advisor: Prof. Vishal M. Patel

Secondary Reader: Prof. Rama Chellappa

Acknowledgments

I am forever grateful for the support and encouragement I received from many individuals over the course of my doctoral studies. First and foremost, I extend my gratitude to my advisor, Professor Vishal Patel, and consider myself fortunate to have been guided him. I am grateful to his scientific insights, wisdom, and support over the years. I would like to thank Professors Carey Priebe, Wei Shen, Shinji Watanabe for serving on my GBO committee. I would like to thank Professor Carlos Castillo, for serving on my proposal committee. I am grateful to Professor Alan Yuille, for serving on both my GBO and dissertation committee. I am also grateful to Professor Rama Chellappa, for serving on both my proposal and dissertation committee.

I am thankful to all the exceptional colleagues and friends I made while working at the VIU-Lab. For a few specific mentions: I would like to thank Pramuditha for helping and guiding me in my research work, especially on the anomaly/novelty detection; Rajeev, I will always remember the discussions we had on both research and cricket; Jose, Vibashan, and Shao-Yuan, some of the excellent juniors from our lab with whom I was fortunate to get a chance to work on exciting new projects. Special thanks to Jose and Vibashan for tolerating me both as a labmate and a roommate. I would also like to

thank Vishwanath, who has been an amazing mentor, friend, and ultimate coffee-buddy throughout my doctoral journey.

Lastly, I am forever indebted to my parents for their love and support all through my life. Also, I am grateful to all family members for always supporting me all through my journey. Especially to my brother and cousins, who have been cheering for my success all these years.

Dedicated to my father Bharat, my mother Alka, and my brother Dhairav for all the love and support granted over years.

Table of Contents

Abstract ii

Acknowledgments iv

Dedications vi

Table of Contents vii

List of Tables xiii

List of Figures xv

1 Introduction 1

1.1 Unknown instance detection 2

1.1.1 One-class problem setting 4

1.1.2 Multi-class problem setting 6

1.2 Domain shift 6

1.3 Data privacy 9

1.4 Outline 9

2 Related Work **12**

2.1 One-class classification 12

2.2 Multi-class novelty detection 15

2.3 User active authentication 16

2.4 Domain adaptation 17

2.4.1 Multi-class novelty detection under domain shift . . . 18

2.4.2 Object detection under adverse-weather conditions . . 19

2.5 Federated learning 20

3 Background **22**

3.1 One-class classification 22

3.1.1 One-class Support Vector Machines (OC-SVM) 22

3.1.2 Support Vector Data Descriptor (SVDD) 24

3.2 Domain adversarial training 26

3.3 Object detection 29

3.3.1 Faster-RCNN 30

3.3.2 Single Shot Multi-Box Detector 32

4 One-class Convolution Neural Networks **34**

4.1 Motivation 34

4.2 Proposed approach 35

4.2.1 Feature extractor 35

4.2.2 Classification network 36

4.2.3 Loss function 36
4.3 Experimental results 37
4.3.1 Abnormality detection 39
4.3.2 User active authentication 39
4.3.3 Novelty detection 40
4.4 Results and discussion 41
4.4.1 Conclusion 43

5 Auto-encoder Regularized One-class CNNs **44**
5.1 Face-based active authentication 44
5.2 Proposed approach 46
5.2.1 Feature extractor 47
5.2.2 Classification network 48
5.2.3 Decoder network 48
5.2.4 Loss functions 48
5.3 Experimental results 50
5.3.1 Datasets 52
5.3.2 Qualitative evaluation 53
5.3.3 Quantitative evaluation 55
5.3.4 Conclusion 57

6 Utilizing Patch-level Activity Patterns for Multi-class Novelty Detection **59**

6.1 Patch-level activities of a recognition model 59
6.2 Proposed method 61
6.2.1 Global inference network 62
6.2.2 Local inference network 62
6.2.3 Novelty detection network 63
6.2.4 Leveraging a reference dataset 65
6.3 Experiments and results 66
6.3.1 Novelty detection datasets 66
6.3.2 Training details 68
6.3.3 Network architecture 68
6.3.4 Quantitative analysis 69
6.3.4.1 Novelty detection performance 69
6.3.4.2 Ablation analysis 73
6.3.5 Qualitative analysis 75
6.3.5.1 Fine-tune baseline vs proposed method . . . 75
6.3.6 Conclusion 79

7 Multi-class Novelty Detection under Distribution Shift 80

7.1 Motivation 80
7.2 Robust novelty detection under distribution shift 83
7.2.1 Problem setting 83
7.2.2 Simple approaches 84

7.2.3 Proposed method 87
7.3 Experiments and results 90
7.3.0.1 Digits: SVHN, USPS, MNIST 93
7.3.0.2 Office31 : Amazon, Webcam, DSLR 95
7.3.1 Conclusion 96

8 Prior-based Domain Adaptive Object Detection **98**
8.1 Motivation 98
8.2 Proposed method 100
8.2.1 Detection network 101
8.2.2 Prior-adversarial training 103
8.2.2.1 Haze prior 105
8.2.2.2 Rain prior 106
8.2.3 Residual Feature Recovery Block (RFRB) 107
8.2.4 Overall loss 108
8.3 Experiments and results 108
8.3.1 Implementation details 108
8.3.2 Adaptation to hazy conditions 110
8.3.3 Adaptation to rainy conditions 115
8.3.4 Conclusion 118

9 Federated Learning-based User Authentication **119**
9.1 Federated average vs non-IID data 119

9.2 Federated active authentication 122
9.2.1 Proposed training methodology 122
9.2.2 Testing 124
9.3 Experiments and results 125
9.3.1 Implementation details 125
9.3.2 Datasets 126
9.3.3 Experiments 127
9.3.4 Fedarated/split learning vs proposed method 130
9.3.5 Conclusion 131

10 Conclusion and Future Work **133**

10.1 Future research directions 134

Bibliography **136**

List of Tables

4.1 Performance comparison between proposed OC-CNN method and existing one-class classification methods with AlexNet backbone 38

4.2 Performance comparison between proposed OC-CNN method and existing one-class classification methods with VGG16 backbone 38

5.1 Ablation study results for the proposed OC-ACNN method . 55

5.2 Performance comparison between proposed OC-ACNN method and existing one-class classification methods with AlexNet backbone 55

5.3 Performance comparison between proposed OC-ACNN method and existing one-class classification methods with VGG16 backbone 55

5.4 Performance comparison between proposed OC-ACNN method and existing one-class classification methods with VGGFace backbone 56

6.1 Novelty detection network architecture details 67

6.2 Quantitative analysis of multi-class novelty detection approach with existing methods in the literature measured using area under the receiver operating characteristic curve 69

6.3 Ablation study of multi-class novelty detection method 74

7.1 Domain adaptive multi-class novelty detection performance on digits benchmark datasets 94

7.2 Domain adaptive multi-class novelty detection performance on Office31 benchmark dataset 96

8.1 Performance comparison for the Cityscapes → Foggy-Cityscapes experiment 112

8.2 Performance comparison for the Cityscapes → RTTS experiment 114

8.3 Quantitative analysis of the adaptation experiments from WIDER-Face to UFDD Haze and Rain 114

8.4 Performance comparison for the Cityscapes → Rainy-Cityscapes experiment 115

9.1 Performance comparison between one-class based authentication model and federated learning based model 127

9.2 Impact of number of unauthorized user on average detection accuracy 129

List of Figures

1.1 Closed-set vs. open-set recognition 3

1.2 One-class vs multi-class problem setting 4

1.3 Typical example of multi-class novelty detection scenario . . . 5

1.4 Illustration of detection performance under dataset distribution shift 7

1.5 Quantitative performance analysis of object detector models . 8

2.1 A graphical illustration of popular statistical one-class classification methods 13

3.1 Domain adaptation by backpropagation for classification task 28

3.2 Illustration of popular detection frameworks 30

4.1 One-class Convolution Network block diagram 36

4.2 Sample images of one-class classification datasets for abnormality detection, user authentication, novel font-style detection . 39

5.1 An overview of a typical AA system 45

5.2 An overview of the proposed OC-ACNN method 47

5.3 Qualitative evaluation of OC-ACNN 54

6.1 Grad-cam visualizations based on labels 60

6.2 Grad-cam visualizations for known and novel categories . . . 60

6.3 Local-Global network training diagram for multi-class novelty detection 64

6.4 Patch-level activities captured by known and novel categories 76

6.5 Failure cases of the multi-class novelty detection method . . . 77

6.6 Qualitative examples of patch-level activities for wrongly identified novel classes for Caltech-256 dataset 78

7.1 Overview of multiple-class novelty detection under dataset distribution shift 81

7.2 Digits example: Novelty detection performance reduction under distribution shift 82

7.3 Domain adaptive multi-class methods block diagram 85

7.4 Sample images of domain adaptive multi-class novelty detection benchmark datasets 91

8.1 Weather conditions modeled using clean image and weather-specific prior 99

8.2 Block diagram of prior based domain adaptive detection method 102

8.3 Qualitative detection results on hazy images 112

8.4 Detection results on Rainy-Cityscapes 117

9.1 Performance of federated averaging under non-IID conditions 120

9.2 Block diagram describing the training of the proposed method for federated active authentication 121

9.3 Toy example with three users to show the effectiveness of the proposed method compared to one-class modeling based methods 123

9.4 Sample face images from face authentication benchmark datasets 126

9.5 Comparing the performance with federated averaging and split learning 130

Chapter 1

Introduction

The success of deep learning has been greatly beneficial for various fields such as natural language processing [130], [19], [9], robotics [76], [116], [134], computer vision [58], [53], [40], etc. This is especially evident in the case of computer vision, where majority of the progress can be largely attributed to the advancements in deep convolutional neural networks (DCNN) [58]. Owing to their learning capacity, DCNN models have achieved state-of-the-art performance in many vision tasks such as object classification ([40], [45], [43]), semantic segmentation ([69], [144], [15]), and object detection ([102], [101], [67]). This has led to DCNN's increased popularity in several real world applications as compared to the classical computer vision techniques. However, these systems suffer from three major issues, the problem of 1) unknown instance detection, 2) dataset distribution shift, and 3) data privacy. The first problem of unknown instance detection stems from the fact that existing visual recognition models are trained with the assumption that label set during training and testing are exactly same. This limits the model into identifying any unknown categories as known, which can result in erroneous

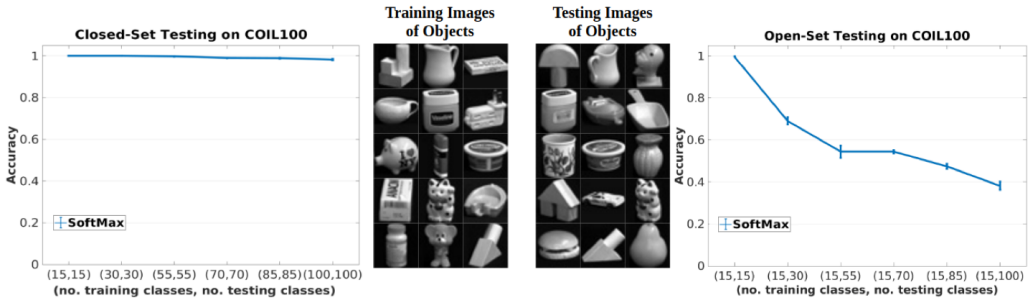
predictions when deployed in real-world scenario. The second problem of dataset distribution shift is caused due to the dynamic nature of the real-world visual data. For example, data collected in one weather conditions are visually distinct from other weather conditions (clean weather vs hazy weather). Hence, a visual recognition model trained on supervised data of one condition does not generalize well to other conditions, and results in reduced performance. Lastly, the third problem stems from the fact that all existing visual recognition models are trained on a central server assuming all the data is available at once in a single location. However, this is not possible in many cases where data is distributed across multiple locations and can not be shared with a central server. For example, in the case of device user authentication sending user data from phone to a server would violate user privacy. Hence, in such cases the model training at central server would not be possible and would need to devise a training strategy that can adjust to these challenges.

Hence, addressing these problems is of utmost importance for real-world deployment of any visual recognition system. In what follows, we discuss each of these problems in detail.

1.1 Unknown instance detection

As discussed earlier, the recent advancements in computer vision have resulted in significant improvements specifically for the classification task [40], [123]. The rise of deep convolutional neural network has resulted in error rates surpassing the human-level performance [39]. These promising results, enable their potential use in many real world applications. However, when

deployed in a real world scenario, such systems are likely to observe samples from classes not seen during training. Since, the traditional training methods follow this closed-set assumption, the classification systems observing any unknown class samples are forced to recognize it as one of the known classes. As a result, it affects the performance of these systems. The Fig. 1.1 illustrates this with the example of unknown instance detection for COIL100 dataset [80]. Here, the full dataset is divided into 15 known categories and 85 unknown categories. A deep neural network based visual recognition model is trained on only the known categories and is tested on both known and unknown categories. We threshold the softmax scores of the visual recognition model to predict if a test sample belongs to known category or unknown category. As we can see from the Fig. 1.1, the visual recognition model drops in performance as we increase the number of unknown categories in the test set. This shows the effect a closed-set training strategy has on the visual recognition system when tested with unknown categories.



(a) (b) (c) (d)

Figure 1.1: Closed-set vs. open-set recognition. (a) Closed-set testing results on the COIL100 dataset. (b) Sample images used for training the algorithm. (c) Samples images used for testing the algorithm (also includes instances from unknown categories). (d) Open-set testing results on the COIL100 dataset.

Hence, it becomes critical to correctly identify test samples as either known

or novel/unknown for a visual recognition model. The problem of unknown instance detection is studied in two different settings, namely, 1) one-class and 2) multi-class. In the one-class problem setting, the task is to learn a visual recognition model from data available of a single category, and during testing the model should be able to identify any unknown categories from the given single category. In the multi-class problem setting, the task is to learn a visual recognition model from a dataset having multiple categories, and the task is to enclose all the categories as one and detect any test samples coming from any unknown categories, i.e., categories that were not present in the training data. In the following sections, we will discuss these problem settings in detail.

1.1.1 One-class problem setting

Multi-class classification entails classifying an unknown object sample into one of many pre-defined object categories. In contrast, in one-class classification,

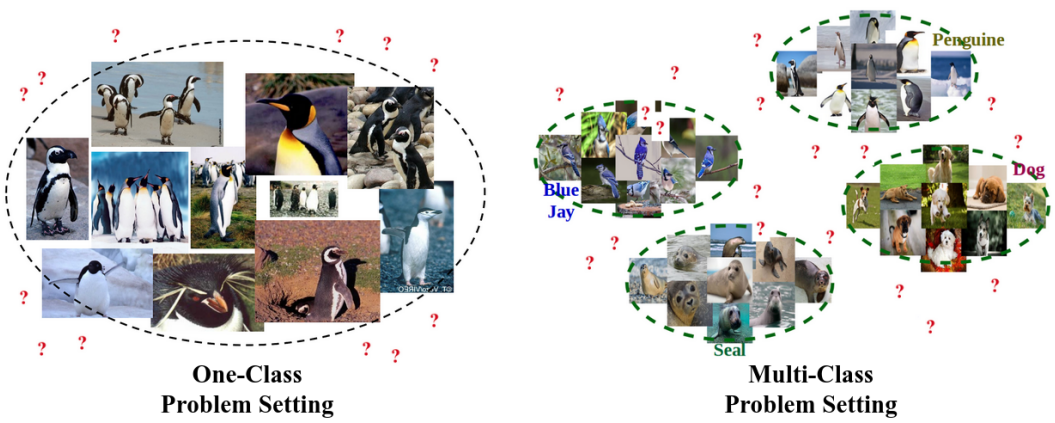


Figure 1.2: The figure illustrates how one-class setting differs from a multi-class setting. Specifically, in the one-class classification setting the model has to learn from data available from only one category. Here, unknown instances are indicated as question mark, as we don't have it available during training.

the objective is to identify objects of a particular class (also known as positive class data or target class data) among all possible objects by learning a classifier from a training set consisting of only the target class data. The absence of data from the negative class(es) makes the one-class classification problem difficult. This difference is also illustrated in Fig. 1.2 with an example of animal classification task. As we can see from the Fig. 1.2, in the case of one-class problem setting during training the data from only one category (e.g. penguin) is available, whereas in the case of multi-class setting data from multiple categories (e.g. penguin, seal, dog, blue jay) is available.

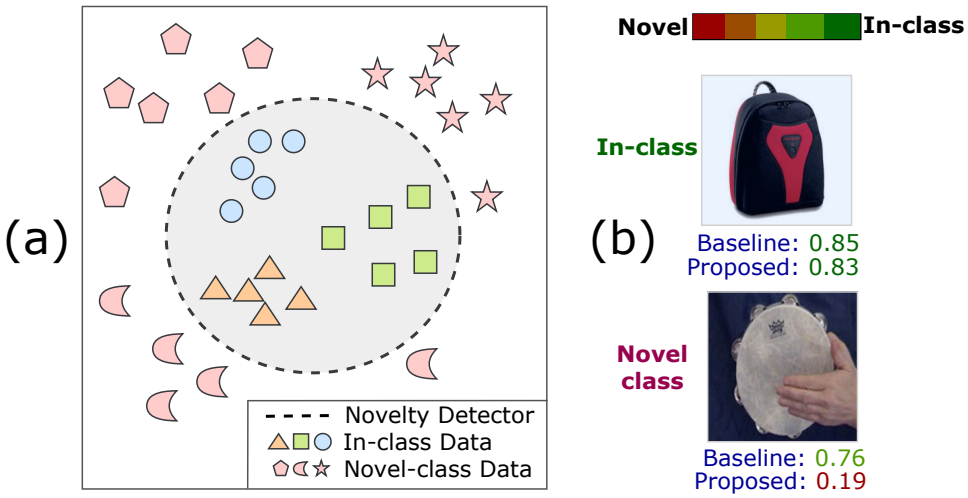


Figure 1.3: (a) Typical example of a multiple class novelty detection scenario, where a novelty detector is used to differentiate between in-class and novel class data. (b) Baseline and the proposed method are able to produce high scores for in-class data. However, for novel class data the proposed approach does is better at assigning low scores compared to the baseline. Here, the "Baseline" refers to the novelty detection using traditional deep convolutional neural network with penultimate layer scores.

1.1.2 Multi-class problem setting

Unlike one-class, in the case of multi-class problem setting there are more than one known categories to learn from [7, 65, 83, 97]. The goal in the case of multi-class problem setting is then to detect data samples of unknown categories instances and improve the robustness for the model against such cases. This is useful in many real-world vision applications. For example, in the case of autonomous navigation systems, it is important to stop and re-plan the navigation path by detecting an object as unknown category rather than wrongly classifying it and risking a potential crash. This is illustrated in Fig. 1.4(a) with a toy example. The known category data have three categories, namely, blue-circle, orange-triangle, and green-square. The goal would be then to learn a decision boundary enclosing these categories shown in the figure as novelty detector which can identify any test sample coming from categories like pentagon, half-moon, or star categories as unknown by giving it a low score as shown in Fig. 1.4(b).

1.2 Domain shift

Most DCNN-based models need to be trained in a supervised fashion, which has been made possible due to the availability of large datasets having thousands of images annotated with ground-truth labels [18], [25], [64]. As discussed earlier, one of the major drawbacks is the poor generalization capability of DCNN models to visually distinct images compared to the training images. For instance, a detection model trained with a dataset collected in Rome may

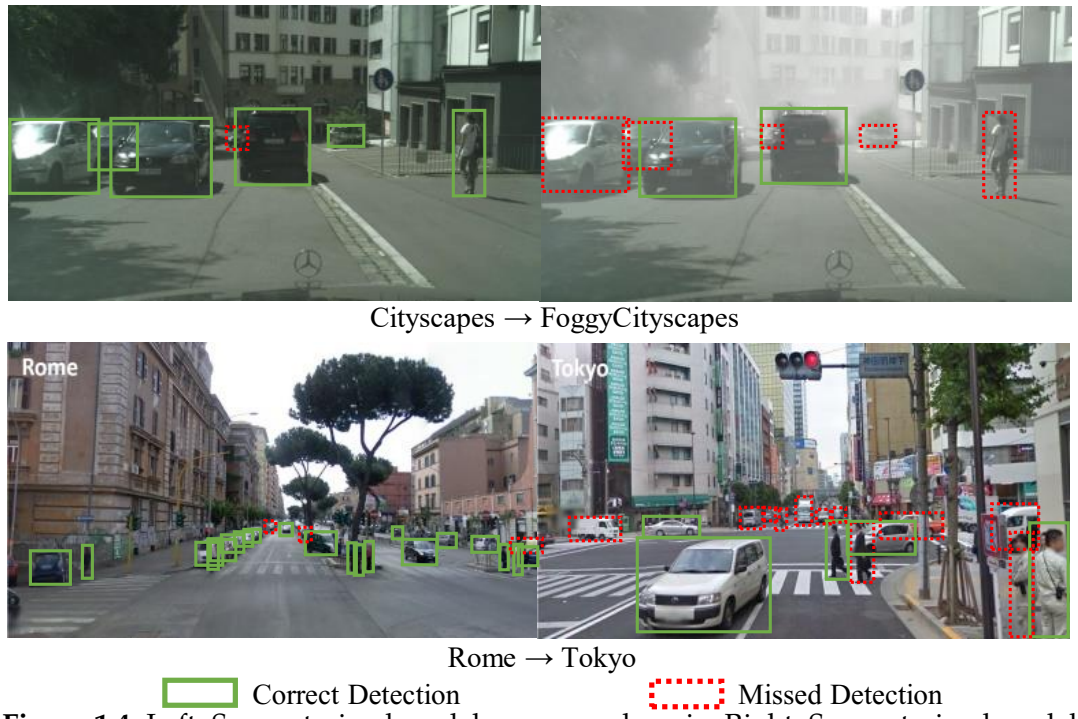


Figure 1.4: Left: Source trained model on source domain, Right: Source trained model on the target domain. Top row: A detection model trained on the Cityscapes dataset, when evaluated on the FoggyCityscapes dataset, it fails to detect cars and pedestrians due to the domain shift caused by fog. Bottom row: A model trained in Rome, when evaluated on another city such as Tokyo, performs poorly due to differences in scene appearances, weather, objects, etc. These examples show that the detection models generalize poorly under the domain shift.

not necessarily perform well on images from Tokyo due to the changes in the appearance of scenes/objects and/or weather between them, as illustrated in bottom row of Fig. 1.4. A similar example is shown for cases such as sunny to foggy weather in Fig. 1.4 top row. Fig. 1.5 shows quantitatively the performance drop of different deep learning based object detectors that are trained on one particular dataset, when evaluated on different datasets. This problem where models, trained on one particular dataset (also known as source dataset), do not generalize well to a dataset that has a different distribution (also known as target dataset) is commonly referred to as *domain*

shift or *distribution shift* in the literature [4], [92], [31].

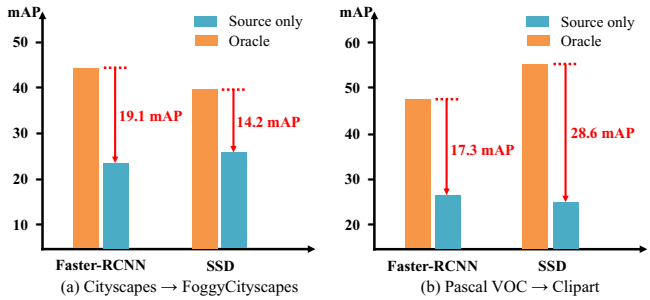


Figure 1.5: Illustration of detector performance; In (a), the model is trained on Cityscapes and evaluated on FoggyCityscapes and in (b), the model is trained on Pascal VOC and evaluated on Clipart. We can observe a significant drop in the performance of the detector when there is a distribution shift in the training and test data.

A straightforward approach to solve this distributional shift problem is annotating the target dataset images with ground-truth detection labels. However, this might prove to be infeasible considering that the labor cost of the annotation process is prohibitively expensive for all visually distinct conditions. To circumvent this issue, many methods rely on the principles of unsupervised domain adaptation [4], [92], [86] which involves training the DCNN model with both labeled source dataset and unlabeled target dataset having visually distinct appearance.

In this thesis, we study this problem of domain adaptation for the task of multi-class novelty detection and object detection under adverse weather conditions, respectively in Chapter 6 and Chapter 8.

1.3 Data privacy

As we discussed earlier, traditional deep network training assumes that all training data are available at a single data center location for training. Furthermore, these data centers may not allow a direct sharing of their data due to privacy concerns. Federated learning [75] and split learning [36] frameworks were specifically proposed to address these issues. Federated learning enables such decentralized deep network training by effectively combining models trained by the individual data centers in a central server [75]. Additionally, such decentralized training protects the privacy of data at individual data centers. This enables a safe collaboration among the data centers to learn a better deep network model without sacrificing user privacy. We will study this issue for the case of user authentication model in Chapter 9.

1.4 Outline

The rest of this thesis is organized into the following chapters:

In Chapter 2, we discuss existing works in the field of one-class classification, multi-class novelty detection, domain adaptative object detection, federated learning etc.

In Chapter 3, we briefly discussed concepts related to one-class classification, domain adaptation, federated learning etc.

In Chapter 4, we present a strategy to train a convolutional neural network in an end-to-end manner for a one-class classification problem setting. We show the benefits of such training compared to just utilizing pre-trained

features and off-the-shelf one-class classifiers like OC-SVM and SVDD.

In Chapter 5, we extend the one-class convolutional neural network proposed in the Chapter 4 by adding a regularization constraint and show its effectiveness in one of the one-class classification application of face-based active user authentication.

In Chapter 6, we study the identification of unknown instances in the multi-class classification setting. Specifically, we explore the use of patch-level activity patterns to identify unknown/novel category instances during testing.

In Chapter 7, we consider the domain shift problem for the multi-class novelty detection task. We study the behavior of existing novelty detection methods under the dataset distribution shift and propose different techniques to mitigate the domain gap issue.

In Chapter 8, we study the domain shift problem for the case of general object detection task. Specifically, we target the use of domain-specific prior information to aid the domain adaptation training for generalizing object detection models to unlabeled data degraded by adverse weather conditions like rain and haze.

In Chapter 9, we tackle the issue of data privacy and decentralized training for the case of face-based user active authentication. Specifically, we devise a training strategy that can learn an authentication model from multiple user/client-devices connected to a central server, without sharing any user data with the central server.

Finally, we conclude the thesis in the Chapter 10. We briefly discuss the takeaways from the thesis. Also, we present potential directions for the future

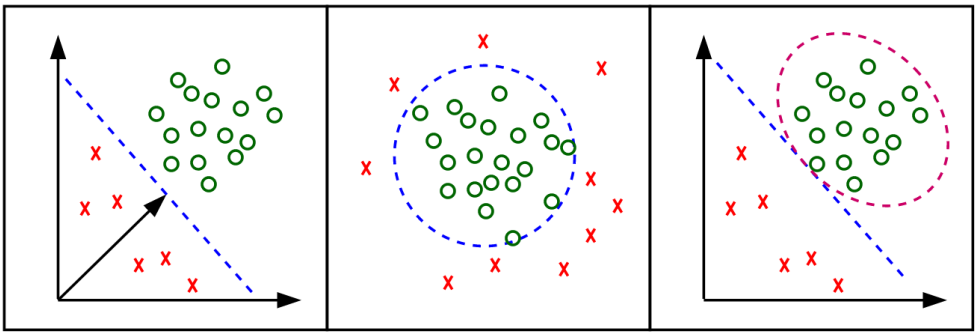
research on the topics discussed in the thesis.

Chapter 2

Related Work

2.1 One-class classification

Various methods have been proposed in the literature for one-class classification [93]. In particular, many of the one-class classification methods are based on the Support Vector Machines (SVM) formulation [115], [74], [23]. SVMs are based on the concept of finding a boundary that maximizes the margin between two classes and are shown to work well for binary and multi-class classification. However, in one-class problems the infromation regarding the negative class data is unavailable. To deal with this issue, Scholkopf *et al.* [113] proposed one-class SVM (OC-SVM), which tackles the absence of negative class data by maximizing the boundary with respect to the origin. Another popular approach inspired by the SVM formulation is Support Vector Data Description (SVDD) introduced by Tax *et al.* [124], in which a hypersphere that encloses the target class data is sought. Various extensions of OC-SVM and SVDD have been proposed in the literature over the years. Another



(a) OC-SVM (b) SVDD (c) Single-MPM

Figure 2.1: A graphical illustration of popular statistical one-class classification methods. Green circles show the target class data, red crosses show the unknown data (i.e. anomaly, novelty, outlier etc.), blue doted lines/circles show the decision boundaries captured by the respective methods. Pink dotted line in Fig. 2.1(c) shows the boundary of zero error of probability. (a) OC-SVM, maximizing the margin of a hyperplane with respect to the origin. (b) SVDD, finding a hypersphere that encloses the given data. (c) MPM, finding a hyperplane that minimizes the misclassification probability.

approach for one-class classification is based on the Minimax Probability Machines (MPM) formulation [56]. Single class MPM [32], [95] seeks to find a hyper-plane similar to that of OC-SVM by taking second order statistics of data into consideration. Hence, single class MPM learns a decision boundary that generalizes well to the underlying data distribution. Fig. 2.1 presents a high-level overview of different one-class classification methods. Though, these approaches are powerful tools in identifying the decision boundary for target data, their performance depends on the features used to represent the target class data.

In recent years, several attempts have been made to counter the problem of training a neural network for one-class classification [105], [104], [14], [57], [100], [146], [13], [96]. These approaches can be broadly classified in to two categories, generative approaches [100], [146], [13] and discriminative

approaches [14], [96]. Generative approaches use generative frameworks such as auto-encoders or Generative Adversarial Networks (GAN) [33] for one-class classification. For example, Ravanbakhsh *et al.* [100] and Sabokrou *et al.* [104] proposed deep auto-encoder networks for event anomaly detection in surveillance videos. However, in their approaches the focus is mainly on the image-level one-class classification. Work by Lawson *et al.* [57] developed a GAN-based approach for abnormality detection. Sabokrou *et al.* [105] extended that idea for detecting outliers from image data using an auto-encoder based generator with adversarial training. In general, these generative models such as GANs are very difficult to train as compared to the discriminative classification networks [111].

Compared to the generative approaches, discriminative approaches for one-class classification have not been well explored in the literature. One such approach by Perera and Patel [96] utilize an external reference dataset as the negative class to train a deep network for one-class classification using a novel loss function. In contrast to this method, we do not make use of any negative class data in our approach. In another approach, Chalapathy *et al.* [14] proposed a novel SVM inspired loss function to train a neural network for anomaly detection. With some inspirations from other statistical approaches for one-class classification (i.e. taking origin as a reference to find the decision boundary), we propose a novel method called, One-Class CNN (OC-CNN), to learn representations for one-class problems with CNNs trained end-to-end in a discriminative manner.

2.2 Multi-class novelty detection

Some of the earlier methods proposed for multi-class novelty detection include [7, 65], which utilize use the feature encoding of in-class data to learn a subspace (referred to as null space of training data in [7, 65]), and during inference the novelty score is calculated based on the distance of a test sample projected onto that subspace with the learned in-class data projections. However, these methods can not be integrated with deep convoluitonal neural networks (DCNN) to perform end-to-end training.

Over the years many novelty detection methods have been proposed some the earliest methods include principle component analysis-based [42, 128], support vector machine-based [114, 124], sparse representation-based [133, 138], nearest neighbors-based [24, 37, 52]. In some of the recent works, Bodesheim *et al.* [7] proposed a kernel-based method that projects all in-class data onto a subspace (referred to as null-space of training data), where all in-class categories are forced to have zero intra-class variance. Specifically, they employ a special case of linear discriminant analysis formulation, called Null-space Foley-Shannon Transform (NFST), to achieve zero intra-class variance. The smallest distance between the test sample projection with the class projections is used to decide whether an input is from a known class or a novel class. Liu *et al.* [65] pointed out that NFST training does not scale well with the increase in dataset size due to its high computation cost. To counter that, they proposed an incremental addition of classes to learn NFST subspace, which results in improved scalability with increased dataset size. Bodesheim *et al.* [6] proposed another variant of NFST-based novelty detection method which

rather than using all in-class data samples, learns the NFST model based on the k nearest neighbor samples. This selective sampling helps to locate the local manifold on the feature space and learn specific models for each test sample.

However, all of these methods provide a general framework for novelty detection and none of them are specifically designed for DCNNs. Schultheiss *et al.* [117] proposed a DCNN-based novelty detection method by examining the extreme signatures observed in the penultimate layer. More precisely, depending on the input data there are specific dimensions in the penultimate layer of DCNNs, which produce high activation values (referred to as extreme value signatures) if the input is from novel class. Recently, Perera *et al.* [97] proposed a DCNN-based training method using a reference dataset. Instead of just utilizing pre-trained models trained on some reference dataset, they propose to use samples from the reference dataset as well. They show that having access to these additional data samples acts as a novel class proxy and benefits the novelty detection aspect of DCNNs. The reference dataset used during training, enables learning of negative filters which forces low activations at penultimate layer, when the input data is not from a novel class.

2.3 User active authentication

In this thesis, we also focus on face-based user active authentication which is one of the most useful applications of unknown user detection.

User active authentication has been considered by many works in the

literature [95], [94]. Most conventional approaches utilize off-the-shelf one-class classification models such as one-class support vector machine (OC-SVM) [113], support vector data descriptor (SVDD) [124], mini-max probability machine (MPM) [56] etc. These one-class classifiers are trained on either hand-crafted features or features extracted from a pre-trained deep neural network. Few recent works attempt to extend these basic one-class classifier formulations by adding more constraints to their objective functions. Noticeably, the work by Perera and Patel [95] extends single mini-max probability machine (SMPM) formulation [32] with additional hyperplane constraint to propose a better one-class classifier called dual-minimax probability machines (DMPM). Many works have explored the use of different biometric modalities such as touch patterns, keystrokes, voice, face for user authentication [20], [55], [28], [120]. More recent works have focused on face-based authentication systems [95], [94].

2.4 Domain adaptation

The domain shift problem has been well-studied in the literature for the image classification task. It is studied specifically in the context of unsupervised domain adaptation. In unsupervised domain adaptation, it is assumed that images in the source dataset are available with category labels, while no label information is provided for the target images. The most popular approaches for this task are based on CNNs. Some of these approaches include feature distribution alignment [129], [30], [122], [108], similarity learning [98], residual transfer [70], [71], and generative adversarial network-based methods [44],

[78], [41], [112]. These methods mostly consider a setting where both source and target datasets have equal number of categories and also consider the classification task. Unlike these methods, we consider the task of multi-class novelty detection and object detection in adverse weather conditions. For each of these tasks there exist a few similar works in the literature.

2.4.1 Multi-class novelty detection under domain shift

Some works have started to consider different settings where the number of categories in the source dataset and the target dataset are not the same. These extensions include partial domain adaptation [11], universal domain adaptation [136] and open-set domain adaptation [87]. Partial domain adaptation assumes that target domain categories are a subset of the source domain categories and hence only a part of the source dataset is useful during adaptation. Whereas open-set domain adaptation assumes that the source domain categories are a subset of the target domain categories and hence only a part of the target data is useful for the adaptation. Universal domain adaptation brings both open-set and partial settings together into a single framework. All of these modifications to the original domain adaptation problem setting are designed to improve the domain adaptation performance on more practical scenarios. The most related problem to the proposed scenario available in the literature is open-set domain adaptation proposed by Busto and Gall *et al.* [87]. However, we would like to point out that there are some key differences between open-set domain adaptation and the proposed approach. Specifically, in open-set domain adaptation, the target categories are a superset

of the source categories, i.e., there are some unknown categories available in the target dataset. Since, no labels are provided for the target domain, the challenge for open-set domain adaption method is to separate out the samples belonging to known and unknown categories in the available target dataset. This extends the domain adaptation capability to a real-world scenario where the target category set will be a superset of the source. In the domain adaptive multi-class novelty detection problem, we do not modify the domain adaptation setting like the open-set domain adaptation, but on the contrary, utilize the domain adaptation techniques to improve generalization of novelty detection methods on different data domains. Specifically, in the case of domain adaptive multi-class novelty detection, we have labeled data from the source domain and unlabeled data from the target domain and both of these domains share the same category set. Also, unlike open-set domain adaptation, where unknown category data samples are accessible during training, here, unknown category data samples are only observed during testing. The end goal for the proposed problem is to utilize the source domain information to create a better novelty detection model for the target domain data. Since both methods follow different problem settings, either of the methods would not be optimal for the other problem setting.

2.4.2 Object detection under adverse-weather conditions

Earlier methods considering adaptation of object detector models for adverse weather conditions include [16, 49, 107, 121]. Specifically, Chen *et al.* [16] assumed that the adversarial weather conditions result in domain shift, and

they overcome this by proposing a domain adaptive Faster-RCNN approach that tackles domain shift on image-level and instance-level. Following the similar argument of domain shift, Shan *et al.* [121] proposed to perform joint adaptation at image level using the Cycle-GAN framework [147] and at feature level using conventional domain adaptation losses. Saito *et al.* [107] proposed to perform strong alignment of the local features and weak alignment of the global features. Kim *et al.* [49] diversified the labeled data, followed by adversarial learning with the help of multi-domain discriminators. Cai *et al.* [10] addressed this problem in the semi-supervised setting using mean teacher framework. Zhu *et al.* [148] proposed region mining and region-level alignment in order to correctly align the source and target features. Roychowdhury *et al.* [103] adapted detectors to a new domain assuming availability of large number of video data from the target domain. These video data are used to generate pseudo-labels for the target set, which are further employed to train the network. Most recently, Khodabandeh *et al.* [48] formulated the domain adaptation training with noisy labels. Specifically, the model is trained on the target domain using a set of noisy bounding boxes that are obtained by a detection model trained only in the source domain.

2.5 Federated learning

Federated Averaging (FedAvg) is one of the most widely used federated learning algorithm to train deep network models [75], [77]. In FedAvg, a model is initialized at a central server and sent to all data centers, which then train their individual models with their locally available data. These

local models are then sent back to the central server, where all local models' parameters are averaged to create a global model. This global model is then again sent back to the individual data centers for another round of local training and the process is repeated until the global model converges.

Split learning also enables training of deep network when data is shared across multiple devices. Gupta *et al.* [36] first introduced split learning where, the whole deep model is divided into two parts. The first part remains on the local device and the second part is kept on the server. The whole model is trained through backpropagation by passing gradient information from server to local devices. Vepakomma *et al.* [131] and Poirot *et al.* [99] utilized split learning framework to train a deep model network on patient data from multiple institutions without having to share the raw patient data. Additionally, Thapa *et al.* [125] proposed an approach that utilizes the principles of both FL and SL to create a fusion method for distributed learning.

Chapter 3

Background

3.1 One-class classification

3.1.1 One-class Support Vector Machines (OC-SVM)

One-class SVM is a special case of Support Vector Machine (SVM) formulation. In a binary SVM, the hyper-plane that separates the two classes with that largest possible margin is found. The hyper-plane is defined by support vectors. In the case of one-class classification, there exists only positively labeled data during training. In One-class SVM (OCSVM), hyperplane corresponding to negative class are set to be the origin of the coordinate system [113]. Therefore, the objective of OCSVM boils down to finding a hyper-plane furthest away from the origin, where positively labeled data exists in the positive half space of the hyper-plane. When this constraint is relaxed using slack variables, the optimization objective can be written as:

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\xi},b} \quad & \tfrac{1}{2}||\mathbf{w}||^2 + \tfrac{1}{\nu N}\textstyle\sum_i \xi_i - b \\ \text{s.t.} \quad & \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle \geq b - \xi_i, \xi_i \geq 0, \end{aligned} \tag{3.1}$$

where, the column vector $\boldsymbol{\xi} = [\xi_i,\ \xi_2,\ \ldots,\ \xi_N]$ and each ξ_i is the slack variable corresponding to the i^{th} training sample (i.e. vectorized image), Φ is a mapping function that maps $\mathbf{x}_i$ to a kernel space where dot products are defined using a kernel function $K(\cdot,\cdot)$, b is the bias term and ν is a trade-off parameter, and N is number of training examples. When the optimization is solved, inference on a query sample $\mathbf{x}_{test}$ can be done using the condition $\text{sgn}(\langle \mathbf{w}, \phi(\mathbf{x}) \rangle - b)$.

Eq. 3.1 can be modified with the help of Lagrange multipliers $\alpha_i, \beta_i \geq 0$ as follows:

$$\mathcal{L}(\mathbf{w}, \boldsymbol{\xi}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}||\mathbf{w}||^2 + \frac{1}{\nu N}\sum_i \xi_i - b - \sum_i \alpha_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle - b + \xi_i) - \sum_i \beta_i \xi_i, \tag{3.2}$$

where the column vectors $\boldsymbol{\alpha} = [\alpha_i,\ \alpha_2,\ \ldots,\ \alpha_N]^T$ and $\boldsymbol{\beta} = [\beta_i,\ \beta_2,\ \ldots,\ \beta_N]^T$. Setting derivatives with respect to primal variables to zero, it can be shown that $\mathbf{w} = \sum_i \langle \alpha_i, \Phi(\mathbf{x}_i) \rangle$, $\alpha_i = \frac{1}{\nu N} - \beta_i \leq \frac{1}{\nu N}$ and $\sum_i \alpha_i = 1$. By substituting these values in Eq. 3.1, the dual optimization problem can be derived as:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \tfrac{1}{2}\textstyle\sum_i \sum_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq \tfrac{1}{\nu N}, \textstyle\sum_i \alpha_i = 1. \end{aligned} \tag{3.3}$$

Furthermore, it can be shown that when $0 \leq \alpha_i \leq \frac{1}{\nu N}$ is satisfied the bias term can also be expressed as:

$$b = \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle = \sum_j \alpha_j K(\mathbf{x}_i, \mathbf{x}_j). \tag{3.4}$$

With the dual form of the problem, as shown in Eq. 3.3, the optimal values of parameters in problem shown in Eq. 3.1 can be found using the kernel

function $K(\cdot,\cdot)$ without explicitly defining the mapping function $\Phi(\cdot)$. The decision for any test image x_{test} that is vectorized as $\mathbf{x}_{test}$ can also be expressed in terms of the kernel function using the dual variables and vectorized training images as follows:

$$\text{sgn}(\sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_{test}) - b), \tag{3.5}$$

3.1.2 Support Vector Data Descriptor (SVDD)

The SVDD [124] formulation closely follows the OCSVM objective. However, instead of learning a hyper-plane to separate data from origin, Tax *et al.* [124] propose to find the smallest hyper-sphere that can fit given training samples. The hyper-sphere is characterized by its mean vector (or centroid of hyper-sphere) $\mathbf{o}$ and radii $r_d > 0$. The volume of hyper-sphere is minimized by minimizing $r_d \in \mathbb{R}$ while making sure hyper-sphere encloses most of the training samples. This objective can be written down in the form of optimization problem as:

$$\begin{array}{ll} \min\limits_{\mathbf{o},\boldsymbol{\xi},r_d} & r_d^2 + \lambda \sum_i \xi_i \\ \text{s.t.} & \|\mathbf{x}_i - \mathbf{o}\|^2 \leq r_d^2 + \xi_i, \xi_i \geq 0 \; \forall i. \end{array} \tag{3.6}$$

Parameter λ controls the trade-off between errors and the objective. Similar to the OCSVM, the above objective can be modified with the help of the Lagrangian multipliers and the updated optimization problem can be re-formulated as:

$$\mathcal{L}(r_d, \mathbf{o}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{,}) = r_d^2 + \lambda \sum_i \xi_i - \sum_i \alpha_i (r_d^2 + \xi_i - (\|\mathbf{x}_i\|^2 - 2\langle \mathbf{o}, \mathbf{x}_i \rangle + \|\mathbf{o}\|^2)) - \sum_i \gamma_i \xi_i, \tag{3.7}$$

where, the column vectors $\boldsymbol{\alpha} = [\alpha_i,\ \alpha_2,\ \ldots,\ \alpha_N]^T$ and $\boldsymbol{\gamma} = [\gamma_i,\ \gamma_2,\ \ldots,\ \gamma_N]^T$. By setting derivatives of primal variables to zero, it can be shown that $\sum_i \alpha_i = 1$, $\mathbf{o} = \sum_i \alpha_i \mathbf{x}_i$ and $\lambda - \alpha_i - \gamma_i = 0$. By substituting to Equation 3.7, the dual form can be obtained as:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \textstyle\sum_i \sum_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \sum_i \alpha_i \langle \mathbf{x}_i, \mathbf{x}_i \rangle \\ & \text{s.t. } 0 \leq \alpha_i \leq \lambda, \textstyle\sum_i \alpha_i = 1. \end{aligned} \tag{3.8}$$

A given test sample $\mathbf{x}_{test}$, is assigned a positive label if it is inside the identified hyper-sphere. More precisely, if the following condition is met:

$$\|\mathbf{x}_{test} - \mathbf{o}\|^2 = \langle \mathbf{x}_{test}, \mathbf{x} \rangle - 2 \sum_i \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle + \sum_i \sum_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \leq r_d^2. \tag{3.9}$$

Since, the dual form and the inference equation both include inner product terms of $\mathbf{x}_i$ and $\mathbf{x}$, SVDD can be extended to a kernel formulation by simply replacing product terms by a kernel function that corresponds to some mapping function Φ as, $\langle \Phi(\mathbf{x}_j), \Phi(\mathbf{x}_i) \rangle = K(\mathbf{x}_i, \mathbf{x}_j)$. The kernalized version of the optimization problem of dual form then can be expressed as:

$$\begin{aligned} \min_{\mathbf{ff}} \quad & \textstyle\sum_i \sum_j \alpha_i \alpha_j (K(\mathbf{x}_i, \mathbf{x}_j) - \sum_i \alpha_i (K(\mathbf{x}_i, \mathbf{x}_i))) \\ & \text{s.t. } 0 \leq \alpha_i \leq C, \textstyle\sum_i \alpha_i = 1. \end{aligned} \tag{3.10}$$

Here, due to $\sum_i \alpha_i = 1$, that in the case where the kernel function only depends on the difference between the two vectors, i.e., when $K(\mathbf{x}_1, \mathbf{x}_2)$ depends only on $\mathbf{x}_1 - \mathbf{x}_2$, the linear term of the dual objective function becomes constant and the optimization becomes equivalent to the dual form of OCSVM in Equation 3.3 discussed in the previous section.

3.2 Domain adversarial training

Let us denote the source dataset as, $\mathcal{S} = \{X_s^i, Y_s^i\}_{i=1}^{N_s}$, and it consists of N_s number of images. Here, X_s^i denotes i^{th} image and Y_s^i denotes the corresponding bounding box annotations with category label. Similarly, let us denote the target dataset as, $\mathcal{T} = \{X_t^i\}_{i=1}^{N_t}$ having N_t number of target domain images with no ground-truth annotations. Ben *et al.* [4] proposed a framework to perform domain adaptation for the given setup, i.e., labeled source dataset and unlabeled target dataset, with theoretical upper bounds on the target performance. Ben *et al.* [4] designed a $\mathcal{H}\Delta\mathcal{H}$-distance to measure the divergence between two sets of samples that have different data distributions, as is the case for the domain adaptation problem. Let us consider an arbitrary source domain image $X_s \in \mathcal{S}$ and an arbitrary target domain image $X_t \in \mathcal{T}$. Furthermore, let us consider a domain discriminator denoted as, $D : X \rightarrow \{0, 1\}$, that takes in any image $X \in \{\mathcal{S} \cup \mathcal{T}\}$ and predicts the domain of the input image. classifies source domain image $X_s \in \mathcal{S}$ as label 0, and target domain image $X_t \in \mathcal{T}$ as label 1. Considering $\mathcal{H}$ to be a set of possible domain discriminators, the $\mathcal{H}\Delta\mathcal{H}$-distance can be defined as follows:

$$
\begin{aligned}
d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) = 2 \sup_{(D, D') \in \mathcal{H}^2} \Big| & \mathop{\mathbf{E}}_{X \sim \mathcal{S}} [D(X) \neq D'(X)] \\
& - \mathop{\mathbf{E}}_{X \sim \mathcal{T}} [D(X) \neq D'(X))] \Big|,
\end{aligned}
\tag{3.11}
$$

where $\mathbf{E}_{X \sim \mathcal{S}}$ and $\mathbf{E}_{X \sim \mathcal{T}}$ denotes the expected domain classification errors over the source and target domain dataset, respectively. More precisely, the Eq. 3.11 measures the divergence by the disagreement of the hypothesis sampled from

$\mathcal{H}$. The ideal joint hypothesis is defined as:

$$D^* = \underset{D \in \mathcal{H}}{\operatorname{argmin}} \operatorname{Err}_{\mathcal{S}}(D^*) + \operatorname{Err}_{\mathcal{T}}(D^*). \tag{3.12}$$

Here, the terms $\operatorname{Err}_{\mathcal{S}}$ and $\operatorname{Err}_{\mathcal{T}}$ denote the expected prediction errors on the source and target domain data samples, respectively. This distance is often used in the domain adaptation literature to measure the adaptability between any give source and target domain datasets. Ben *et al.* [4] present a theorem that further defines the upper bound on the target error as:

$$\begin{aligned} &\forall D \in \mathcal{H},\ \operatorname{Err}_{\mathcal{T}}(D) \leq \\ &\qquad \operatorname{Err}_{\mathcal{S}}(D) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + Const. \end{aligned} \tag{3.13}$$

We can note from the Eq. 3.13, the target error is upper bounded by three terms, namely expected prediction error on the source domain, domain divergence denoted in Eq.3.11, and few constant terms. More details regarding both Eq. 3.11 and Eq. 3.13 are provided in [4]. A majority of the domain adaptation works in the literature rely on this formulation and focus on minimizing the upper bound on the target error by reducing the domain divergence between the source and target domain.

The adversarial feature learning is built on this theory. Specifically, the overall strategy involves minimizing the upper bound given in Eq. 3.13 by directly minimizing the $\mathcal{H}\Delta\mathcal{H}$-distance. As we can notice from $\mathcal{H}\Delta\mathcal{H}$-distance given in Eq. 3.11, this distance is inversely proportional to the error rate of the domain classifier D. The goal in a domain adaptation scenario is to reduce this distance, i.e., increase the domain classifier error. Ganin *et*

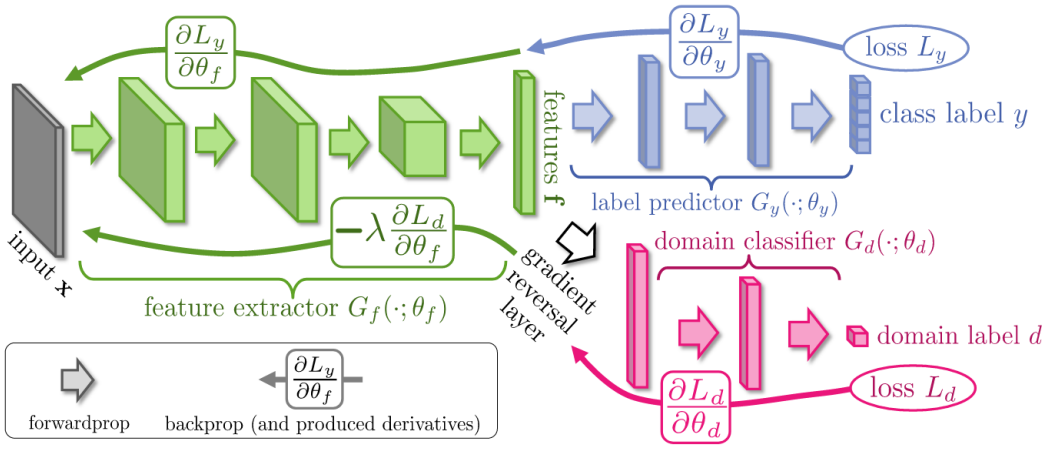


Figure 3.1: Domain adaptation by backpropagation as proposed in [31] with the example of classification task.

al. [31] exploited this and proposed a novel gradient reversal approach to train any neural network model for domain adaptation. The overall goal is to achieve a domain invariant feature space of a backbone neural network with the help of a neural network-based domain classifier. Suppose we denote a domain classifier network as D and the backbone feature extractor network as F. In that case, the feature extractor network also tries to increase the domain classifier loss. The network F tries to minimize the task-specific loss (classification/segmentation/detection loss) and maximize the domain classification loss in the overall training pipeline. The network D is trained to minimize domain classification loss. In addition to the task-specific loss, an additional loss involving domain classification is added. This loss is termed as adversarial loss [31] and it can be written as:

$$\max_{F} \min_{D \in \mathcal{H}} \{\mathbf{E}_{\mathcal{S}}(D) + \mathbf{E}_{\mathcal{T}}(D)\}, \tag{3.14}$$

where $\mathcal{H}$ denotes the hypothesis space for the domain classifier and F is the

feature extractor network. $\mathbf{E}_{\mathcal{S}}(D)$ and $\mathbf{E}_{\mathcal{T}}(D)$ denote the expected domain classification error over source and target domain, respectively. Eq. 3.14 is implemented with the help of a gradient reversal layer which is applied before the input to the domain classifier as shown in Fig. 3.1. The gradient reversal layer during feed-forward acts as an identity function and the gradients are multiplied with -1 during backpropagation. In effect, this forces feature extractor F to maximize the domain classification loss while minimizing the task-specific loss resulting in the domain invariant feature space as proven by Ben *et al.* [4].

3.3 Object detection

Over the years, deep convolutional neural network based object detectors have demonstrated exceptional improvements in performance on a variety of datasets and have become an integral part of various computer vision applications. There are a variety of surveys [66], [145], [149] on the topic covering wide range of techniques proposed over the past decade for object detection. The most popular frameworks for object detection are Faster-RCNN [102], You Only Look Once (YOLO) [101] and Single Shot Multi-box Detector (SSD) [67]. The majority of domain adaptive object detection works are based on the Faster-RCNN and a few others use SSD. Other recent frameworks include, Fully Convolutional One Stage (FCOS) Object Detection [126] and DEtection TRansformer (DETR) [12]. However, these frameworks have been only scarcely used for the domain adaptive object detectors. In what follows, we briefly describe the widely used detection frameworks in the domain

adaptive detection literature.

3.3.1 Faster-RCNN

The Faster-RCNN framework, proposed by Ren *et al.* [102], follows a two-stage object detection approach and it consists of three major components: *1)* a backbone CNN, *2)* a Region Proposal Network (RPN), and *3)* a Region-of-Interest (RoI) based classifier (RCN). Fig. 3.2(a) shows an overview of the Faster-RCNN architecture. Consider a dataset, $\mathcal{D} = \{X^i, Y^i\}_{i=1}^{N}$, having N images, with each image X^i with ground-truth annotation Y^i. Here, the ground-truth annotation Y^i denotes both bounding boxes and respective object categories in the corresponding image X^i. As shown in Fig. 3.2(a), an input image (X^i) is forwarded through the backbone network resulting in a set of feature maps. These feature maps are then fed to the RPN network which generates a set of candidate object proposals. The RPN network relies on pre-defined anchor boxes of multiple sizes and aspect ratios in order to effectively learn to generate the candidate proposals. Subsequently, each proposal is then transformed into fixed-size features using RoI-pooling. Finally, the pooled features are then forwarded through the RCN, which predicts the category

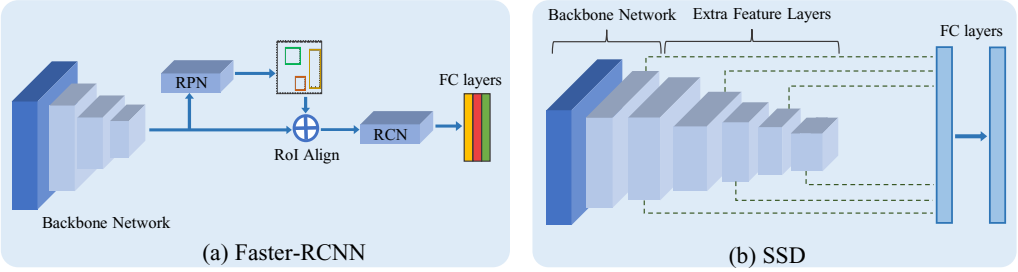


Figure 3.2: Illustration of popular detection frameworks: (a) Faster-RCNN [102], (b) Single Shot Multi-Box Detector (SSD) [67].

label for each candidate proposal in addition to refining its bounding box. For training the RPN candidate, a category-agnostic binary label (of being an object or not) is assigned to each anchor. The j^{th} anchor is assigned a label, denoted as $y_b^j \in \{0, 1\}$, as positive (or 1) if it has the highest Intersection over Union (IoU) overlap with one of the ground-truth boxes or if it has an IoU overlap higher than 0.7 with any of the ground-truth boxes in the corresponding image. Similarly, a negative label (or 0) is assigned to the anchor if IoU ratio is lower than 0.3 for all ground-truth boxes. The RPN is then tasked to perform a binary classification to identify whether the candidate bounding box proposal corresponds to one of the objects in the image and learn the offset between the ground-truth bounding box, denoted as $\mathbf{b}^j$, and respective anchor box to get final bounding box prediction, denoted as $\tilde{\mathbf{b}}^j$. The offset learning is supervised with the help of a regression loss applied on the bounding box parameters. Both these losses are combined together to obtain the final loss for region proposal network as shown below:

$$\begin{aligned} \mathcal{L}_{rpn} \;=\; & \frac{1}{N_b} \sum_j \mathcal{L}_{rpn}^{bce}(y_b^j,\; p_b^j) \\ & + \;\lambda_{rpn} \frac{1}{N_{bbox}} \sum_j p_b^j \mathcal{L}_{rpn}^{reg}(\mathbf{b}^j,\; \tilde{\mathbf{b}}^j), \end{aligned} \tag{3.15}$$

where j is the index of an anchor box in the mini-batch and p_b^j is the probability assigned to the respective anchor box being an object. The loss, $\mathcal{L}_{rpn}^{reg}$, computes the smooth-L1 distance between the given ground truth bounding box and the predicted bounding box $\tilde{\mathbf{b}}^j$.

Next, the RCN network is trained to perform classification of RoI-pooled features using cross entropy loss with $K + 1$ class classification, denoted as $\mathcal{L}_{rcn}^{ce}$.

Here, K denotes the number of categories in the dataset and an additional class to represent the background category. Additionally, the RCN is also tasked to predict the bounding box offset through regression loss similar to the RPN network.

The overall loss function used to train the entire Faster-RCNN network is trained is defined as:

$$\mathcal{L}_{det}^{frcnn} = \mathcal{L}_{rpn} + \mathcal{L}_{rcn}. \tag{3.16}$$

More details regarding the anchor boxes, regression losses, training, and architecture can be found in the [102].

3.3.2 Single Shot Multi-Box Detector

Liu *et al.* [67] proposed a single shot object detection framework which consists of forwarding the image through a single stage as opposed to two stages in the Faster-RCNN detector. Fig. 3.2(b) illustrates the SSD detection architecture. By following this approach, SSD eliminates the need for an object proposal stage, making it simpler and computationally efficient as compared to the Faster-RCNN approach. The SSD framework employs VGG16 as the backbone network which is used for extracting feature map of size $H \times W$ from an input image X. For each feature map location, SSD discretizes the output space of the bounding boxes into a set of default bounding boxes. A convolutional layer is added that for each feature map location predicts a score for a category or offsets relative to the default box coordinates. The set of default boxes contain bounding boxes of multiple pre-defined aspect ratios and scales to match any object shape in the image better. Furthermore, SSD combines predictions from

feature maps at multiple scales to better handle the object scales with respect to the image. Once the model predictions are available, they are matched with the ground-truth box and category to preform an end-to-end training with regression ($\mathcal{L}_{reg}$) and classification ($\mathcal{L}_{cls}$) loss. The final detection loss is a combination of both regression and classification losses and is defined as follows:

$$\mathcal{L}_{det}^{ssd} = \mathcal{L}_{reg} + \mathcal{L}_{cls}. \tag{3.17}$$

In the case where there are no predicted bounding boxes that can be matched with one of the ground-truth bounding boxes, the regression loss is set to zero. More details regarding the default boxes, box matching algorithm bounding box regression losses, training procedure, and architecture details can be found in [67].

Chapter 4

One-class Convolution Neural Networks

4.1 Motivation

One of the existing methods for one-class based unknown instance detection includes Perera and Patel [96], which utilizes an external reference dataset as the negative class to train a deep network for one-class classification using a novel loss function. However, the availability of such a reference dataset is not always guaranteed for one-class classification. Hence, for the method proposed in this chapter, we do not make use of any negative class data or a reference data. More precisely, the proposed method circumvents the issue of unavailability of negative class data by introducing a pseudo-negative category modelled using Gaussian distribution and enables the end-to-end model training of any deep convolution neural network in one-class problem setting. In what follows, we describe the proposed training methodology in detail.

4.2 Proposed approach

Fig. 4.1 gives an overview of the proposed CNN-based approach for one-class classification. The overall network consists of a feature extractor network and a classifier network. The feature extractor network essentially embeds the input target class images into a feature space. The extracted features are then appended with the pseudo-negative class data, generated from a zero centered Gaussian in the feature space. The appended features are then fed into a classification network which is characterized by a fully connected neural network. The classification network assigns a confidence score for each feature representation. The output of the classification network is either 1 or 0. Here, 1 corresponds to the data sample belonging to the target class and 0 corresponds to the data sample belonging to the negative class. The entire nework is trained end-to-end using binary cross-entropy loss.

4.2.1 Feature extractor

Any pre-trained CNN can be used as the feature extractor. We use the pre-trained AlexNet [53] and VGG16 [123] networks by removing the softmax regression layers (i.e. the last layer) from their networks. During training, we freeze the convolution layers and only train the fully-connected layers. Assuming that the extracted features are D-dimensional, the features are appended with the pseudo-negative data generated from a Gaussian, $\mathcal{N}(\bar{\mu}, \sigma^2 \cdot \mathbf{I})$, where σ and $\bar{\mu}$ are the parameters of the Gaussian and $\mathbf{I}$ is a $D \times D$ identity matrix. Here, $\mathcal{N}(\bar{\mu}, \sigma^2 \cdot \mathbf{I})$ can be seen as generating D independent one dimensional gaussian with σ standard deviation.

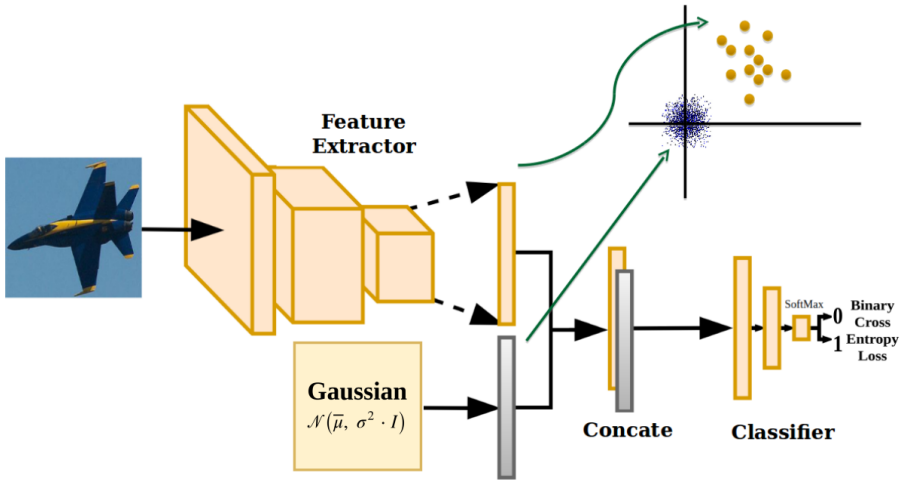


Figure 4.1: Block diagram of the proposed approach. Here, $\bar{\mu}$ and σ are mean and standard deviation parameters of a Gaussian, respectively and **I** is the identity matrix.

4.2.2 Classification network

Due to the appending of the pseudo-negative data with the original features, the classifer network observes the input in the batch size of 2. A simple fully-connected layer followed by a softmax regression layer is used as the classifier network. The dimension of the fully-connected layer is kept the same as the feature dimension. The number of outputs from the softmax layer are set equal to two.

4.2.3 Loss function

The following binary cross-entropy loss function is used to train the entire network

$$L_c = -\frac{1}{2K}\sum_{j=1}^{2K}(y\log(p) + (1-y)\log(1-p)), \tag{4.1}$$

where, $y \in \{0, 1\}$ indicates whether the classifier input corresponds to the feature extractor, (i.e. $y = 0$), or it is sampled from $\mathcal{N}(\bar{\mu}, \sigma^2 \cdot \mathbf{I})$, (i.e. $y = 1$).

Here, p denotes the softmax probability of $y = 0$.

The network is optimized using the Adam optimizer [50] with learning rate of 10^{-4}. The input image batch size of 64 is used in our approach. For all experiments, the parameters $\bar{\mu}$ and σ are set equal to $\bar{0}$ and 0.01, respectively. Instance normalization [22] is used before the classifier network as it was found to be very useful in stabilizing the training procedure.

4.3 Experimental results

We evaluate the performance of the proposed approach on three different one-class classification problems - abnormality detection, face-based user authentication, and novelty detection. Abnormality-1001 [110], UMDAA-02 [73] and FounderType-200 [65] datasets are used to conduct experiments for the abnormality detection, user authentication and novelty detection problems. For all methods compared here, the data is aligned such that objects are at the center with minimal background.

The proposed approach is compared with following one-class classification methods:

- **OC-SVM:** One-Class Support Vector Machine is used as formulated in [115], trained using the AlexNet and VGG16 features.
- **BSVM:** Binary SVM is used where the zero centered Gaussian noise is used as the negative data. AlexNet and VGG16 features extracted from the target class data are used as the positive class data.
- **MPM:** MiniMax Probability Machines are used as formulated in [56].

Table 4.1: Comparison between the proposed and other methods using **AlexNet** as the base network. Results are mean of performance on all classes. Best and the second best performance are highlighted in bold fonts and italics, respectively.

Dataset	OC-SVM	BSVM	MPM	SVDD	OC-NN-lin	OC-NN-sig	OC-NN-relu	OC-CNN	OC-SVM$^+$
Abnormality-1001	0.6057	0.6126	0.5806	0.7873	0.8090	0.6391	0.7372	*0.8264*	**0.8334**
UMDAA-02 Face	0.5746	0.5660	0.5418	0.6448	0.6173	0.6452	0.5943	**0.7017**	*0.6736*
FounderType-200	0.7124	0.7067	0.7085	0.8998	0.8884	0.8696	0.8505	*0.9303*	**0.9350**

Table 4.2: Comparison between proposed and other methods using **VGG16** as the base network. Results are mean of performance on all classes. Best and the second best performance are highlighted in bold fonts and italics, respectively.

Dataset	OC-SVM	BSVM	MPM	SVDD	OC-NN-lin	OC-NN-sig	OC-NN-relu	OC-CNN	OC-SVM$^+$
Abnormality-1001	0.6475	0.6418	0.5909	0.8031	0.7740	0.8373	0.5821	*0.8424*	**0.8460**
UMDAA-02 Face	0.5829	0.5751	0.5473	0.6424	0.6193	0.6200	0.5788	**0.7350**	*0.7230*
FounderType-200	0.7490	0.7067	0.7444	0.8885	0.8986	0.8677	0.8506	*0.9290*	**0.9419**

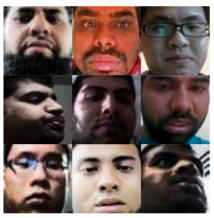
Since, the MPM algorithm involves computing covariance matrix from the data, Principal component analysis (PCA) is used to reduce the dimensionality of the features before computing the covariance matrix.

- **SVDD:** Support Vector Data Description is used as formulated in [124], trained on the AlexNet and VGG16 features.

- **OC-NN:** One-class neural network (OC-NN) is used as formulated in [14]. Here, for fair comparison, instead of using the feature extractor trained using an auto-encoder (as per [14] methodology), AlexNet and VGG16 networks, the same as the proposed method, are used. As described in [14], we evaluate OC-NN using three different activation functions - linear, Sigmoid and ReLU.

- **OC-SVM$^+$:** OCSVM$^+$ is another proposed in this section, where OC-SVM is utilized on top of the features extracted from the network trained using OC-CNN. However, since it uses OC-SVM for classification, it is not end-to-end trainable.

(a) Abnormality-1001



(b) UMDAA-02 Face

(c) FounderType-200

Figure 4.2: Sample images from the datasets used for conducting experiments. (a) Abnormality-1001 (b) UMDAA-02 Face (c) FounderType-200.

4.3.1 Abnormality detection

Abnormality detection (also referred as anomaly detection or outlier rejection) deals with identifying instances that are dissimilar to the target class instances (i.e. abnormal instances). Note that, the abnormal instances are not known a priori and only the normal instances are available during training. Such problem can be addressed by one-class classification algorithms. The Abnormality-1001 dataset [110] is widely used for visual abnormality detection. This dataset consists of 1001 abnormal images belonging to six classes such as Chair, Car, Airplane, Boat, Sofa and Motorbike which have their respective normal classes in the PASCAL VOC dataset [25]. Normal images obtained from the PASCAL VOC dataset are split into train and test sets such that the number of abnormal and normal images in test set are equal. Reported results are averaged for all six classes.

4.3.2 User active authentication

Active authentication refers to the problem of identifying the enrolled user based on his/her biometric data such as face, swipe patterns, and accelerometer patterns [90]. The problem can be viewed as identifying the abnormal user

behaviour to reject the unauthorized user. The active authentication problem has been viewed as one-class classification problem [3]. The UMDAA-02 dataset [73] is widely used dataset for user active authentication on mobile devices. The UMDAA-02 dataset has multiple modalities corresponding to each user such as face, accelerometer, gyroscope, touch gestures, etc. Here, we only use the face data provided in this dataset since face is one of the most commonly used modality for authentication. The face data consists of 33209 face images corresponding to 48 users. As can be seen from this figure, the images contains large variations in pose, illumination, appearance, and occlusions. For each class, train and test sets are created by maintaining 80/20 ratio. Network is trained using the train set of a target user and tested on the test set of the target user against the rest of the user test set data. This process is repeated for all the users and average results are reported.

4.3.3 Novelty detection

The FounderType-200 dataset was introduced for the purpose of novelty detection by Liu et al. in [65]. The FounderType-200 dataset, contains 6763 images from 200 different types of fonts created by the company FounderType. For experiments, first 100 classes are used as the target classes and remaining 100 classes are used as the novel data. The first 100 class data are split into train and test set having equal number of images. For novel data, a novel set is created having 50 images from each of the novel classes. For each class, train set from the known data is used for training the network and known class test set and novel set data are used for evaluation. For example, class

i ($i \in \{1, 2, .., 100\}$) train set is used for training the network. The trained network is then evaluated with class i test set tested against the novel set (containing data of class 101-200). This is repeated for all classes i where, $i \in \{1, 2, .., 100\}$ and average results are reported.

4.4 Results and discussion

The performance is measured using the area under the receiver operating characteristic (ROC) curve (AUROC), most commonly used metric for one-class problems. The results are tabulated in Table 4.2 and Table 4.1 corresponding to the VGG16 and AlexNet networks. AlexNet and VGG16 pretrained features are used to compute the results for OC-SVM, BSVM, SVDD and MPM. The OC-NN results are computed using the linear, sigmoid and relu activations after training on the target class data. The OC-CNN results are computed after training on the target class and for OC-SVM$^+$, an one-class SVM is trained on top of the features extracted from the trained AlexNet/VGG16, and AUROC is computed from the SVM classifier scores.

From the Tables 4.1 and 4.2, it can be observed that either OC-CNN or OC-SVM$^+$ achieves the best performance on all three datasets. MPM and OC-SVM achieve similar performances, while BSVM with Gaussian data as the negative class doesn't work as well. With the BSVM baseline, we show that similar trick we used for proposed algorithm doesn't work well for statistical approaches like SVM. Among the other one-class approaches, OC-NN with linear activation performs the best. However, OC-NN results are inconsistent. For couple of experiments, SVDD was found to be working better than

OC-NN. The reason behind this inconsistent performance can be due to the differences in the evaluation protocol used for OC-NN in [14]. The ratio of the number of target class images to novel/abnormal class images in our evaluation protocol is much higher than the ratio used by Chalpathy et al. [14]. When the ratio is close to one, as is the case for Abnormality-1001 dataset, the OC-NN performs better than SVDD for both AlexNet and VGG16. However, when the ratio is increased (which is more realistic scenario), as is the case for UMDAA-02 and FounderType-200, the performance of OC-NN becomes inconsistent. Whereas, using the proposed approach performs consistently well, providing $\sim$4%, $\sim$10% and $\sim$5% improvements over OC-NN for Abnormality-1001, UMDAA02- Face and FounderType-200 datasets, respectively. Since, the proposed approach is built upon the traditional discriminative learning framework for deep neural networks, it is able to learn better features than OC-NN.

Also as expected, methods based on the VGG16 network work better than the methods based on the AlexNet network. Apart from the FounderType-200 dataset where, OC-CNN with AlexNet works better than VGG16, for all methods VGG16 works better than AlexNet. However, it should be noted that better OC-SVM$^+$ performance for VGG16 indicates that features learned with the proposed approach for VGG16 are better than AlexNet for FounderType-200. Overall, VGG16 gives $\sim$2% improvement over AlexNet.

Another interesting comparison is between OC-SVM and OC-SVM$^+$. OC-SVM uses features extracted from a pre-trained AlexNet/VGG16 network. On the other hand, OC-SVM$^+$ uses features extracted from AlexNet/VGG16

network trained using the proposed approach. OC-SVM$^+$ performs $\sim$18% and $\sim$17% better than OC-SVM on average across all datasets for AlexNet and VGG16, respectively. This result shows the ability of our approach to learn better representations. So, apart from being an end-to-end learnable standalone system, our approach can also be used to extract target class friendly features. Also, using sophisticated classifier has shown to improve the performance over OC-CNN (i.e., OC-SVM$^+$) in majority of cases.

4.4.1 Conclusion

We proposed a new one-class classification method based on CNNs. A pseudo-negative Gaussian data was introduced in the feature space and the network was trained using a binary cross-entropy loss. Apart from being a standalone one-class classification system, the proposed method can also be viewed as good feature extractor for the target class data (i.e. OCSVM$^+$) as well. Furthermore, the consistent performance improvements over all the datasets related to authentication, abnormality and novelty detection showed the ability of our method to work well on a variety of one-class classification applications. In this section, experiments were performed over data with objects centrally aligned. In the future, we will explore the possibility of developing an end-to-end deep one class method that does joint detection and classification.

Chapter 5

Auto-encoder Regularized One-class CNNs

5.1 Face-based active authentication

A simple approach for face-based active authentication (AA) would be to use face images correspoinding to all users and train a system to classify each user in a multi-class fashion. However, such an approach becomes counter-intuitive for AA since it requires the storage of all face images at a centralized location, raising data privacy issues [91]. Hence, one must consider only the data collected from the enrolled user to develop an AA system. In other words, we need to explore possibilities of implementing AA systems using only the user's enrolled data. This motivates us to view AA as a one class classification problem [3]. Fig. 5.1 shows a typical face-based AA system, modelled as a one class classification problem.

Learning a one class classifier based on only the target class data has been one of the most challenging problems in machine learning. Some of the earlier works have used statistical methods to tackle this problem. Such

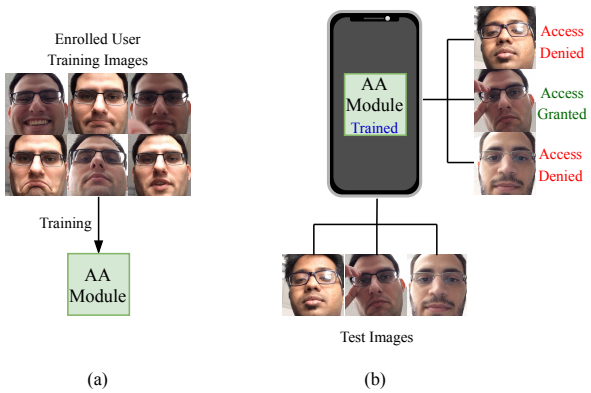


Figure 5.1: An overview of a typical AA system. (a) Data corresponding to the enrolled user are used to train an AA system. (b) During testing, data corresponding to the enrolled user as well as unknown user may be presented to the system. The AA system then grants access to the enrolled user and blocks access to unknown users.

statistical methods usually seek separating hyperplane/hypersphere in the feature space to enclose the target class data [113], [124], [32], [47]. These methods rely on the quality of the representations used for the target class data. Earlier approaches were based on the hand-crafted features. In recent years features based on Deep Convolutional Neural Networks (CNNs) have shown to produce better results than hand-crafted features. Utilizing these powerful feature representations help in learning good decision boundaries, feature representations and classifiers are learned separately. In such a disjoint approach, classification module doesn't influence CNNs to modify the feature representation for a given target class data. Several recent works have explored joint learning of both features and classifiers [96], [14], [84] for one class classification. These methods demonstrated that representation learning together with classifier training results in improved performance. Based on this motivation, an end-to-end learning approach is proposed in this paper which jointly learns feature representations and a classifier for one class

classification. Furthermore, the learned representations are constrained by a decoder network which regularizes the learned representations by enforcing them to reconstruct the original data.

5.2 Proposed approach

An overview of the proposed OC-ACNN network architecture is shown in Fig. 5.2. It consists of three major modules namely, feature extractor network, classification network and decoder network. The feature extractor network generates latent space representations for a given target class data. These latent representations are then fed to a classifier and a decoder network. Before feeding them to the classifier network, they are concatenated with a vector sampled from a zero centered Gaussian $\mathcal{N}(\mu, \sigma \cdot I)$, where σ and μ are the parameters of the Gaussian and I is the identity matrix. This Gaussian vector acts as a pseudo-negative class for the classifier. The classifier network is tasked with discriminating the target class representation from the pseudo-negative Gaussian vector. The decoder network takes in the same latent representation to reconstruct the original input. This enforces the latent representation generated by the feature extractor network to be self-representative i.e., representations are required to generate back the original input images. The classification network and the decoder network are trained end-to-end using a combination of binary cross entropy loss and $L1$ loss, respectively.

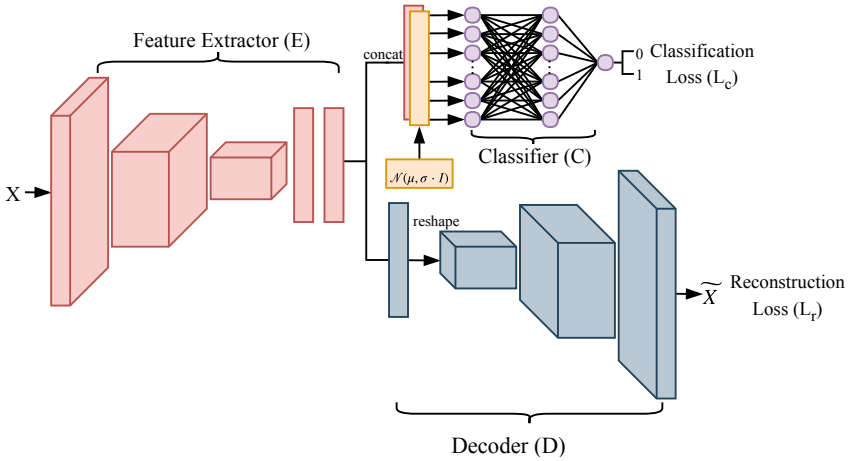


Figure 5.2: Here, X is the input. The feature extractor module (E) can be any pre-trained CNN architecture. Here, AlexNet, VGG16 and VGGFace networks are considered as feature extractor. The decoder module (D) is a simple four layer fully convolutional network. The decoder network essentially reconstructs the input image X, as $\tilde{X}$. The classification network (C) is a fully connected neural network trained to distinguish between feature vectors coming from E and the Gaussian vectors sampled from $\mathcal{N}(\mu, \sigma \cdot I)$. The entire network is trained using a combination of classification loss ($\mathcal{L}_c$) and reconstruction loss ($\mathcal{L}_r$).

5.2.1 Feature extractor

The feature extractor network (E) can be any state of the art network architecture. Here, pre-trained AlexNet [53], VGG16 [123] and VGGFace [89] are considered as feature extractor network. Before using these architectures as feature extractor, the final layer (i.e. softmax regression layer) is removed. While training, we update weights of only the fully connected layers and freeze the weights of convolutional layers. AlexNet and VGG16 utilized here are initialized with the ImageNet pre-trained weights and VGGFace is initialized with the VGGFace dataset pre-trained weights.

5.2.2 Classification network

Assuming that the extracted features are D-dimensional, the features are appended with the pseudo-negative data generated from a Gaussian, $\mathcal{N}(\mu, \sigma \cdot I)$, similar to [85]. Following We use a simple one layer fully connected classifier network (C) with sigmoid activation at the end, as shown in Fig. 5.2. The number of hidden units are the same as the length of the feature vector representation. Because of the Gaussian vector concatenation at the input, the network C observes twice the batch size (N) as of the feature extractor.

5.2.3 Decoder network

The decoder network (D) architecture is a simple four layer fully convolutional neural network. This network takes feature representation learned by the network E and tries to reconstruct the original input. This in effect constraints E to generate representation which have self-representation property. It can be seen as a form of regularization on the feature representation. This regularization can be controlled with parameter λ_r given in Eq. 5.3 of total loss function. Since feature extractor outputs a flattened feature vector, we reshape it to an appropriate size before feeding it to the decoder network. Note that E along with D can be viewed as an auto-encoder network.

5.2.4 Loss functions

The entire network is trained using a combination of two loss functions - classification loss ($\mathcal{L}_c$) and reconstruction loss ($\mathcal{L}_r$). The classification loss is

defined as follows

$$\mathcal{L}_c \;=\; -\frac{1}{2N}\sum_{j=1}^{2N}\left[\, y\cdot\log_2(p)+(1-y)\cdot\log_2(1-p)\,\right], \tag{5.1}$$

where $y \in \{0,1\}$ indicates whether classifier input corresponds to feature extractor (i.e., $y = 1$), or sampled from $\mathcal{N}(\mu, \sigma \cdot I)$, (i.e., $y = 0$). Here, p is the probability of $y = 1$. The classification network C observes twice the input batch size because we append Gaussian vector in batch dimension with extracted features, in Eq. 5.1, the summation is over $2N$.

The $L1$ reconstruction loss is defined as follows

$$\mathcal{L}_r \;=\; \frac{1}{N}\sum_{j=1}^{N} \|X-\tilde{X}\|_1, \tag{5.2}$$

where X and $\tilde{X}$ are the original input image and the reconstructed image, respectively.

Finally, the overall loss function is the sum of $\mathcal{L}_r$ and $\mathcal{L}_c$ defined as follows

$$\mathcal{L}_t \;= \mathcal{L}_c + \lambda_r \mathcal{L}_r, \tag{5.3}$$

where λ_r is a regularization parameter. Furthermore, note that $\tilde{X} = D(E(X))$ and $p = C(E(X))$.

The network is optimized using the Adam optimizer [50] with the learning rate of 10^{-4} and batch size (i.e. N) of 64. For all the experiments, μ, σ and λ_r are set equal to 0.0, 0.01 and 1.0, respectively. The decoder architecture is as follows

ConvTran(1024, 256) - ConvTran(256, 64) - ConvTran(64, 16) - ConvTran(16, 3),

where, ConvTran(*in*, *out*) denotes the transposed convolutions with *in* and *out* as number of input and output feature channels, respectively. All transposed convolutions are used with kernels of size 4×4. ReLU activation is used after every transposed convolution layer except the fourth, where Tanh activation is used. Instance normalization [22] is used before the classifier network and at the end of every transposed convolution layer.

5.3 Experimental results

We evaluate the performance of the proposed approach on three publically available face-based AA datasets – MOBIO [127], UMDAA-01 [26] and UMDAA-02 [73]. The proposed approach is compared with the following one-class classification methods:

- **OC-SVM:** One class SVM as formulated in [113] is used. OCSVM is trained on features extracted from AlexNet, VGG16 and VGGFace.
- **SMPM:** SMPM is used as formulated in [56]. In SMPM formulation, to utilize the second order statistics, covariance matrix computation is required. Hence, before applying SMPM, we reduce the dimensionality of the features extracted from AlexNet, VGG16 and VGGFace using Principle Component Analysis (PCA).
- **SVDD:** Support Vector Data Description is used as formulated in [124], trained on the AlexNet,VGG and VGGFace features.
- **OC-NN:** One-class neural network (OC-NN) is used as formulated in [14]. The encoder network described in [14] is replaced with a pretrained

CNNs, i.e. AlexNet, VGG16 and VGGFace to have a fair comparison between the methods. Apart from this change, the training procedure is exactly the same as given in [14].

The following ablation baselines are also considered to show the contribution of each module in the proposed approach:

- **Auto-Encoder baseline (only $\mathcal{L}_r$):** This is one of the ablation baselines, where we utilize the feature extractor and the decoder networks, and train with only $\mathcal{L}_r$ loss function given in Eq. 5.2. It can also be seen as a generative approach baseline. The reconstruction error is used for classification. In other words, a pre-determined threshold is compared against the reconstruction error and the input is rejected if the error is greater than the threshold. Otherwise, the input is declared as corresponding to the one-class data.

- **Classifier baseline (only $\mathcal{L}_c$):** Another ablation baseline includes using classifier and feature extractor networks trained with only $\mathcal{L}_c$ loss function given in Eq. 5.1. The classification network is not regularized by the decoder network. This baseline is equivalent to the method proposed in [85]. This ablation study will clearly show the significance of using an auto-encoder as a regularizer for one-class classification.

- **Proposed approach OC-ACNN (both $\mathcal{L}_r$ and $\mathcal{L}_r$):** OC-ACNN is the method proposed in this section.

For OC-SVM, SMPM and SVDD distance scores from the hyperplane/hypersphere are used for performance evaluation. For OC-NN, classifier baseline and the

proposed approach, scores from the classifier are used for performance evaluation. As mentioned before, the reconstruction error is used for evaluating the performance of the auto-encoder baseline.

5.3.1 Datasets

MOBIO. The MOBIO dataset is a bi-modal AA dataset containing face images and voice recordings of 150 users. Here, we only consider face images for conducting the experiemnts. For each user the recordings are taken in six sessions at different locations. We combine images from all six sessions. MOBIO contains less variations in pose, illumination etc., as compared to the other datasets. For experiments, first 48 users are considered as target users and the rest are used as unknown users. Target users' data is split into train and test set with 85/15 ratio. For each target user, the training set is used to train the networks. During evaluation, we utilize the test set of the target user along with the data from all unknown users. This process is repeated for all 48 users and average performance is reported.

UMDAA-01 Face. The UMDAA-01 dataset contains face and touch gestures recorded from a mobile device. In total 750 video sequences of 50 users are collected in three different sessions with varying illumination conditions. Data from different sessions are combined for each user and split into train and test sets with 80/20 ratio. Considering one user as the target and the remaining 49 users as unknown, networks are trained using target train set and tested with test set consisting of all 50 users' data (i.e. 1 target and 49 unknown). This experimental protocol is followed for all 50 users and average performance is

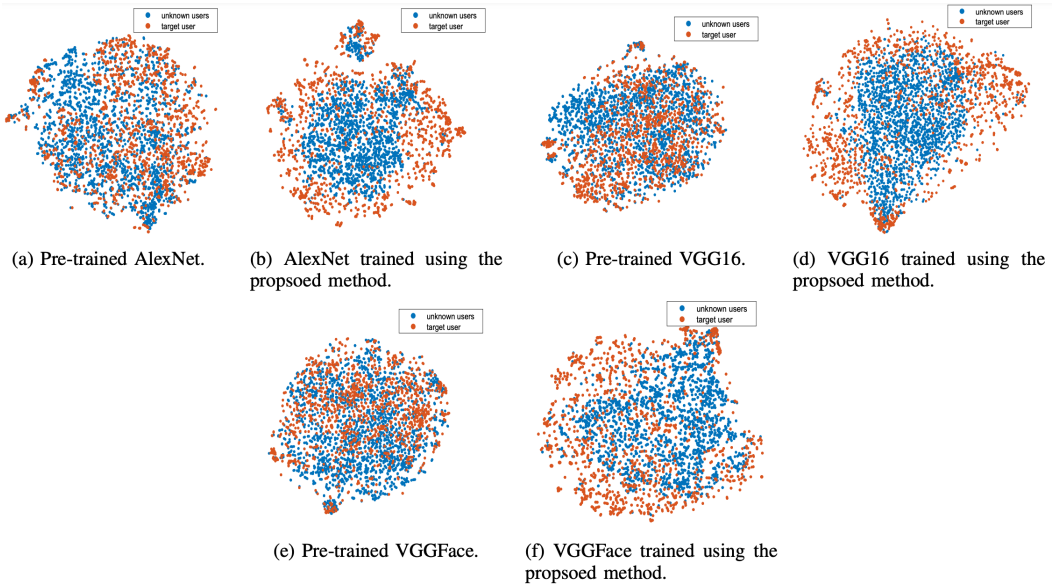
reported.

UMDAA-02 Face. Unlike the above two datasets, the UMDAA-02 dataset has multiple modalities for 44 users e.g. face, gyroscope, swipe patterns, key strokes etc. all recorded from 18 sensors of a Nexus mobile device. Since the dataset was collected over a period of two months, it is an extremely challenging dataset with large variations in pose, illumination, occlusions and other environmental conditions. The number of sessions for each user ranges from 25 to 750 providing large number of frontal face images, i.e. more than 10k images on average per user. For each user, train and test splits are created with 80/20 ratio. We follow similar protocol as that of UMDAA01-Face for all 44 users and report the average performance.

Area Under the ROC curve (AUROC) is used to measure the performance. This is one of the most commonly used metric in the literature for evaluating the performance of the one-class classification methods.

5.3.2 Qualitative evaluation

In this section, we present qualitative evaluation of the proposed approach by comparing the visualizations of feature representations learned by our method with those corresponding to respective pre-trained networks. Fig. 5.3 shows t-SNE [72] visualizations of the feature representations corresponding to AlexNet, VGG16 and VGGFace, respectively. These t-SNE plots are obtained from a single user of the UMDAA-02 Face dataset. Fig. 5.3a, 5.3c and 5.3e show the visualizations corresponding to pre-trained AlexNet, VGG16



(a) Pre-trained AlexNet. (b) AlexNet trained using the propsoed method. (c) Pre-trained VGG16. (d) VGG16 trained using the propsoed method.

(e) Pre-trained VGGFace. (f) VGGFace trained using the propsoed method.

Figure 5.3: t-SNE visualizations of feature representations from the feature Extractor corresponding to a user from the UMDAA-02 Face dataset.

and VGGFace networks, respectively. Fig. 5.3b, 5.3d and 5.3f show the visualizations corresponding to their counterpart networks trained using the proposed approach. As can be seen from these figures, the pre-trained networks generate features that highly overlap between the target and unknown users. This makes sense since these networks are trained using a cross-entropy loss for multi-class classification. As a results, the features from these networks overlap significantly and it makes it difficult for a one-class classifier to correctly identify the separating decision boundary. On the other hand, the feature representations of the same networks trained using the proposed approach are quite distinctive. The learned feature representations corresponding to the target and unknown users are very well separated. These features become extremely useful while identifying the target user against the unknown users, thereby resulting in improved classification performance.

Table 5.1 shows the AUROC results corresponding to each plot for the same user computed using OC-SVM. As can be seen from this table, for all the networks the features learned using the proposed OC-ACNN provides better performance compared to the pre-trained features.

Table 5.1: AUROC results corresponding to the study conducted in Fig. 5.3.

Feature Extractor	Pre-Trained	OC-ACNN
AlexNet	0.5319	0.6780
VGG16	0.5698	0.8194
VGGFace	0.5428	0.8808

Table 5.2: Comparison between the proposed approach and other one-class methods with **AlexNet** as the feature extractor network. Results are the mean of performance on all classes. The performance is measured by AUROC. Best performance is highlighted in bold fonts.

Dataset	OC-SVM	SMPM	SVDD	OC-NN	**Auto-encoder** (only $\mathcal{L}_r$)	**Classifier** (only $\mathcal{L}_c$)	**OC-ACNN** ($\mathcal{L}_c$+$\mathcal{L}_r$)
Mobio	0.6578 ± 0.1132	0.7721 ± 0.1185	0.7851 ± 0.1270	0.7504 ± 0.1512	0.7526 ± 0.1075	0.8191 ± 0.1286	**0.8633 ± 0.1136**
UMDAA-01	0.6584 ± 0.1255	0.7576 ± 0.1149	0.8909 ± 0.0755	0.8684 ± 0.0913	0.6560 ± 0.1066	0.9196 ± 0.0482	**0.9276 ± 0.0465**
UMDAA-02	0.5746 ± 0.0595	0.5418 ± 0.0382	0.6448 ± 0.0725	0.6542 ± 0.0593	0.5952 ± 0.0869	0.7017 ± 0.1007	**0.7398 ± 0.0787**

Table 5.3: Comparison between the proposed approach and other one-class methods with **VGG16** as the feature extractor network. Results are the mean of performance on all classes. The performance is measured by AUROC. Best performance is highlighted in bold fonts.

Dataset	OC-SVM	SMPM	SVDD	OC-NN	**Auto-encoder** (only $\mathcal{L}_r$)	**Classifier** (only $\mathcal{L}_c$)	**OC-ACNN** ($\mathcal{L}_c$+$\mathcal{L}_r$)
Mobio	0.6607 ± 0.1066	0.7266 ± 0.1046	0.8212 ± 0.1130	0.7822 ± 0.1153	0.7457 ± 0.1072	0.8177 ± 0.1132	**0.8705 ± 0.1278**
UMDAA-01	0.6777 ± 0.0946	0.8664 ± 0.0765	0.9011 ± 0.0592	0.8802 ± 0.0976	0.8494 ± 0.0844	0.9348 ± 0.0384	**0.9486 ± 0.0336**
UMDAA-02	0.5828 ± 0.0757	0.5473 ± 0.0447	0.6424 ± 0.0677	0.6199 ± 0.0693	0.6042 ± 0.0939	0.7349 ± 0.0845	**0.8457 ± 0.0581**

5.3.3 Quantitative evaluation

Tables 5.2, 5.3 and 5.4 show the performance on all three datasets based on AlexNet, VGG16 and VGGFace as feature extractors, respectively. The performance of other methods is inconsistent across the experiments. SMPM was found to perform better than OCSVM, while SVDD achieves better performance in many cases beating OC-NN. This may be due to the evaluation

Table 5.4: Comparison between the proposed approach and other one-class methods with **VGGFace** as the feature extractor network. Results are the mean of performance on all classes. The performance is measured by AUROC. Best performance is highlighted in bold fonts.

Dataset	OC-SVM	SMPM	SVDD	OC-NN	**Auto-encoder** (only $\mathcal{L}_r$)	**Classifier** (only $\mathcal{L}_c$)	**OC-ACNN** ($\mathcal{L}_c$+$\mathcal{L}_r$)
Mobio	0.6702 ± 0.1268	0.6619 ± 0.1068	0.7975 ± 0.1250	0.7673 ± 0.1380	0.7339 ± 0.1095	0.8347 ± 0.1324	**0.8859 ± 0.1042**
UMDAA-01	0.6763 ± 0.1237	0.7334 ± 0.1241	0.8745 ± 0.0794	0.8257 ± 0.1381	0.8237 ± 0.0923	0.9432 ± 0.0654	**0.9772 ± 0.0213**
UMDAA-02	0.5712 ± 0.0644	0.5671 ± 0.0597	0.5898 ± 0.0647	0.5987 ± 0.0652	0.6343 ± 0.0723	0.6393 ± 0.0946	**0.8946 ± 0.0535**

protocol difference compared to the one proposed in OC-NN [14]. In OC-NN evaluation protocol, the number of unknown classes used during evaluation are much less than the number of unknown classes used for evaluation (i.e., MOBIO(96), UMDAA-01 Face(49) and UMDAA-02 Face(43)). This can be a reason for the poor performance from OC-NN as compared to SVDD. OC-NN however, manages to perform better than SMPM and OCSVM, and in couple of cases SVDD. Meanwhile the proposed approach achieves superior performance across all the datasets and for different feature extractor models.

Comparing the performance across models, VGGFace outperforms both VGG16 and AlexNet models. This makes sense since face images (i.e. VGGFace dataset) were used to train the original VGGFace model and the corresponding weights are better suited for face-based AA application considered here. In contrast, the VGG16 and AlexNet networks were trained using general object dataset (i.e. ImageNet dataset) for object recognition task. The highest performance for all these networks is achieved for UMDAA-01 Face, since this dataset only contains illumination variations. Though MOBIO contains least variations in image samples, it has large number of unknown classes to compare against. While UMDAA-02 is the most difficult dataset since it contains very unconstrained face images. As a result, the performance on this dataset is lower than the other two datasets. In summary, the proposed

approach observes improvement of ∼6%, ∼9% and ∼16% on average across all datasets corresponding to AlexNet, VGG16 and VGGFace, respectively.

Comparing ablation baselines, the auto-encoder baseline using only the reconstruction loss performs the poorest, while only the classifier baseline performs reasonably well. Auto-encoder and classifier baselines can be categorized as generative and discriminative approach, respectively. Since the discriminative approach can learn better representation, it helps the classifier baseline to improve its performance. However, when the decoder is added to the classification pipeline to regularize the learned representations, it improves the overall performance by ∼6%. This clearly shows the significance of enforcing the self-representation constraints to regularize the learned feature representations for one-class classification.

5.3.4 Conclusion

We proposed a new approach for single user AA based on auto-encoder regularized CNNs. Feature representations are jointly learned with classifier influencing the generated representations. A pseudo-negative Gaussian vector was utilized to train the fully connected classification network. Decoder was introduced to regularize the generated feature representation by enforcing it to be self-representative. Experiments were conducted using the AlexNet, VGG16 and VGGFace networks, which showed the adaptability of the proposed method to work with different types of network architectures. Ablation study was conducted to show the importance of both classification

loss and feature regularization. Moreover, visualizations of the learned representations showed the ability of the proposed approach to learn distinctive features for one-class classification. Furthermore, the consistent performance improvements over all the datasets related to AA showed the significance of the proposed one-class classification method.

Chapter 6

Utilizing Patch-level Activity Patterns for Multi-class Novelty Detection

6.1 Patch-level activities of a recognition model

Deep convolutional neural networks have the ability to learn high-quality representations that are class-discriminative, making them the most successful tool for image recognition. These representations are learned by an end-to-end training and are computed by aggregating patch-level convolution responses (or activation maps) through non-linear activation functions and pooling process. Furthermore, these activation maps are aggregated depending on the strength of the activation to predict the probability scores for each class. The classes are ranked based on the predicted probability score and the class having the maximum score (i.e. rank-1 class) is predicted as the label. Fig. 6.1 illustrates this point with grad-cam [119] visualizations of top-3 classes. Here, the classes are ranked based on the predicted probability scores. The visualizations in Fig. 6.1 are not limited to top-3 classes and can be shown for all

categories in the training set. This figure shows that given an image, a DCNN produces activation maps that has some contribution from all known classes.

For novel class test samples, none of the predictions would be correct, since the training set did not contain these classes. Furthermore, as shown

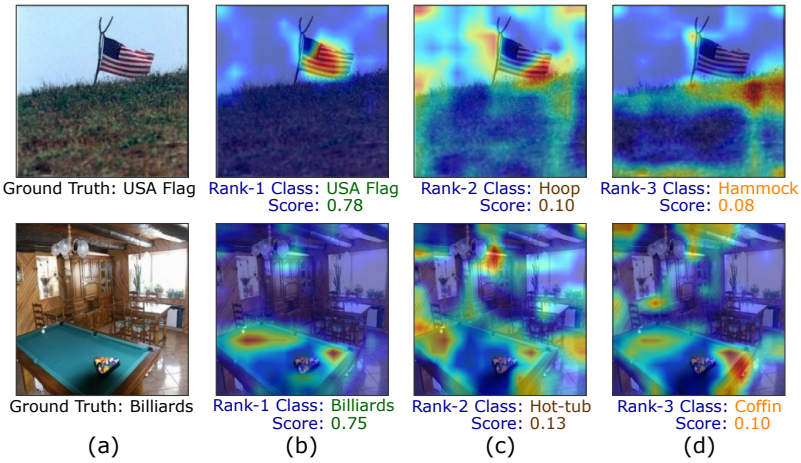


Figure 6.1: (a) Original image with corresponding ground truth label. (b), (c) and (d) represent grad-cam visualizations for rank-1, rank-2 and rank-3 classes and predicted probability scores.

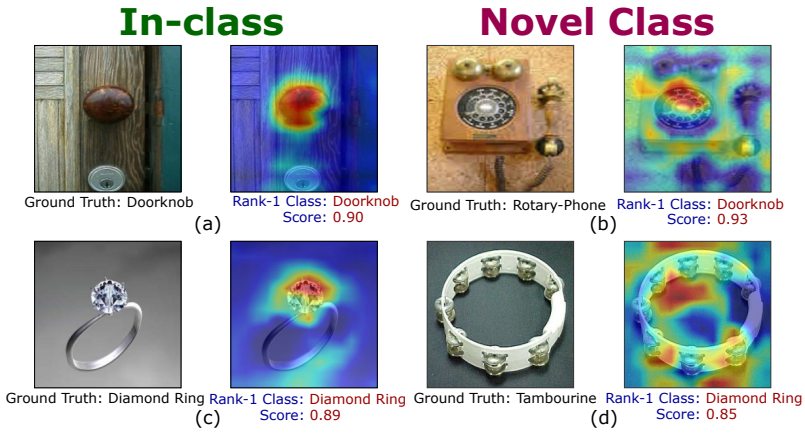


Figure 6.2: (a)-(b) In-class samples from Doorknob and Diamond Ring classes with grad-cam visualizations and the predicted scores. (c)-(d) Novel class samples from Rotary-Phone and Tambourine are mis-classified as Doorknob and Diamond Ring as shown with grad-cam visualizations and predicted scores.

in Fig. 6.2, often the rank-1 prediction scores for novel classes are very high, making it difficult for DCNNs to identify them as novel. However, looking at the examples shown in Fig. 6.2, one can notice that the patch-level activation patterns for both known class samples and novel class samples are different, even when both images are classified as the same class with high scores. The activation patterns for in-class (i.e. known class) samples are focused on the underlying object, whereas for novel class data the patterns are spread out across the image producing high activations at multiple image-patch locations. Given this information, we make an assumption that this type of discrepancy in the patch-level activation pattern exists across all novel class samples. Based on this assumption, we propose a novelty detection algorithm that learns to detect novel class samples by identifying discrepancy in the patch-level activation patterns.

6.2 Proposed method

Let us first consider a multi-class novelty detection problem setup. Here, we have access to only in-class data samples, $\{x_i, y_i\}_{i=1}^{i=n}$, where $y_i \in \{1, 2, ..., K\}$ is the class label corresponding to the data point x_i, n is the total number of data samples and K is the total number of classes. In the following subsections, we provide details of the each individual components of the proposed novelty detection method.

6.2.1 Global inference network

The global inference network can be decomposed in to two parts, feature extractor ($\mathcal{G}$) and classifier ($\mathcal{C}$). The feature extractor ($\mathcal{G}$), processes the image through stacked convolutional, pooling and activation layers to produce a global feature encoding of the object present in the image, as shown in Fig. 6.3(a). The classifier ($\mathcal{C}$), uses this global feature encoding to classify the image into one of K classes. The cross entropy loss used to train such network can be defined as follows

$$\mathcal{L}_{global} = \frac{1}{n}\sum_{i=1}^{n} L_{ce}(\mathcal{C}(\mathcal{G}(x_i)),\, y_i), \tag{6.1}$$

where y_i is the ground truth class label for the input x_i, n is total number of images from known classes and $\mathcal{C}(\mathcal{G}(x_i))$ is the predicted probability vector.

6.2.2 Local inference network

For local inference, the network needs to process individual image patches and provide predictions at patch-level as opposed to the global inference network where the predictions are provided at the image level. To achieve this, we utilize a recently proposed BagNet architecture [8] as local inference network. Specifically, BagNet processes the input image using a series of convolutional layers with 1×1 convolutions and 3×3 convolutions. The limiting of receptive field size restricts the network to perform patch-level processing and produce patch-level feature encodings. These patch-level encodings are used to produce patch-level prediction scores for all K classes, here referred to as local feature encodings. All these predictions are average

pooled to produce the final prediction score, which is trained using the cross entropy loss in an end-to-end fashion. This process is illustrated in Fig. 6.3(b). The local feature encodings provide us with information regarding what each image-patch corresponds to and also the details regarding patch-level activation patterns for a particular class. This information is particularly useful in our approach and is utilized in the next section to train the novelty detection network. The local inference network is trained using the following loss function

$$\mathcal{L}_{local} = \frac{1}{n}\sum_{i=1}^{n} L_{ce}(gap(\mathcal{R}(x_i)),\ y_i), \tag{6.2}$$

where $\mathcal{R}$ denotes the local inference network, $\mathcal{R}(x_i)$ denotes the prediction map having all patch-level prediction scores corresponding to all K classes and gap represents global average pooling operation along the height and width of the prediction map (shown in Fig. 6.3).

6.2.3 Novelty detection network

The proposed novelty detection method utilizing global and local inference is illustrated in Fig. 6.3(c). As discussed earlier, the proposed approach relies on two assumptions, *1)* the activation patterns for a particular global predictions are different in the case of in-class sample and novel class sample, and *2)* for each image from in-class data belonging to a particular class (y_i), DCNN produces activation maps that has some contribution from all known classes.

Based on these assumptions, we train the novelty detection network to model the probability of mis-match (discrepancy) between the predicted label by the global inference and corresponding patch-level activation patterns

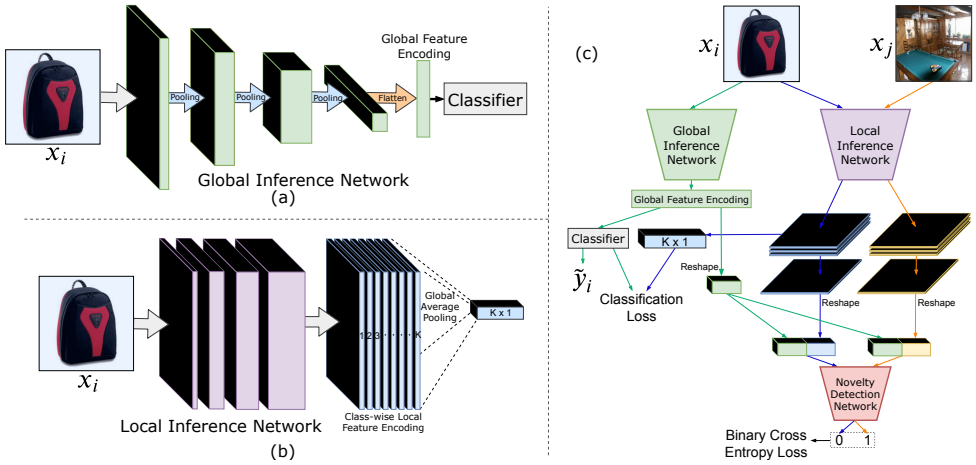


Figure 6.3: (a) The global inference network processes the image to produce a global feature encoding which is used by the classifier to predict the class label. (b) The local inference network architecture provides patch-level features which are used to produce class-wise local feature encoding for all K classes, providing information regarding the presence of all classes at the patch-level. (c) Both global and local network information are combined in a novel training strategy for novelty detection, specifically to model mis-match between local activations and global predictions. For given any image x_i, the global and local features of the predicted class $\tilde{y}_i$ are concatenated to create a positive example. Local feature of the predicted class $\tilde{y}_i$ for another randomly sampled image x_j from a different class is combined with the same global feature to create a negative example. The novelty detection network is trained to distinguish between these positive and negative examples. The Global and Local inference networks are trained using the cross entropy classification loss on their respective predictions. *Note that, both x_i and x_j are sampled from in-class data.*

predicted by the local inference. This modeling should help during testing to detect novel samples by detecting the mis-match between the activation patterns and the prediction. Specifically, consider two randomly sampled images x_i and x_j having corresponding labels y_i and y_j, such that $y_i \neq y_j$. The predicted label and global feature encoding for image x_i is denoted as $\tilde{y}_i = \arg\max_i \mathcal{C}(\mathcal{G}(x_i))$ and $g_i = \mathcal{G}(x_i)$, respectively. The local feature encoding belonging to the predicted class $\tilde{y}_i$ for both images x_i and x_j are denoted as $r_i = \mathcal{R}(x_i)_{\tilde{y}_i}$ and $r_j = \mathcal{R}(x_j)_{\tilde{y}_i}$, respectively. This process is illustrated

in Fig. 6.3(c). The following loss is used for training the novelty detection network

$$\mathcal{L}_{novelty} = \frac{1}{n} \sum_{\substack{i=1, y_i \neq y_j \\ j \sim \{1,..,n\}}}^{n} L_{ce}(\mathcal{N}(cat(g_i, r_i)),\ 0) + L_{ce}(\mathcal{N}(cat(g_i, r_j)),\ 1), \tag{6.3}$$

where $\mathcal{N}$ denotes the novelty detection network and cat represents reshape and concatenation operations. Also, $j \sim \{1, .., n\}$ and $y_i \neq y_j$ denote that for every training image x_i an index j is randomly sampled from the given in-class data, such that both x_j and x_i have different labels. During, testing the novel samples are identified by using predictions from network $\mathcal{N}$. The overall objective for the proposed approach can be written by combining Eq. (6.1)-(6.3) as follows

$$\min_{\mathcal{N},\, \mathcal{G},\, \mathcal{R},\, \mathcal{C}} \ \mathcal{L}_{global} + \mathcal{L}_{local} + \mathcal{L}_{novelty}. \tag{6.4}$$

Details regarding the network architectures and training procedures are provided in supplementary material.

6.2.4 Leveraging a reference dataset

The proposed method can be easily extended in the case where the reference dataset is available. We apply regularization on penultimate activations of the global inference network, similar to the loss function proposed in [21]. Such regularization of the final layer activations penalizes the high activations of any input from the reference dataset. Let us denote the reference dataset as $\mathcal{D}_{ref}$ having m number of images, then the regularization loss can be expressed

as follows

$$\mathcal{L}_{reg} = \frac{1}{m} \sum_{x \in \mathcal{D}_{ref}} \|\mathcal{C}(\mathcal{G}(x))\|_2. \tag{6.5}$$

The final objective function in this case is updated by adding $\mathcal{L}_{reg}$, in Eq. 6.4 as,

$$\min_{\mathcal{N},\, \mathcal{G},\, \mathcal{R},\, \mathcal{C}} \quad \mathcal{L}_{global} + \mathcal{L}_{local} + \mathcal{L}_{novelty} + \lambda \mathcal{L}_{reg}. \tag{6.6}$$

Here, the parameter λ controls the effect of regularization on the final activations, and is chosen using the validation accuracy of the dataset. In experiments, we set parameter λ equal to 0.001.

6.3 Experiments and results

6.3.1 Novelty detection datasets

Caltech-256. The Caltech-256 dataset contains 256 object classes and a total of 30607 images. The dataset has a minimum of 80 images to a maximum of 827 images per category. Based on the protocol defined in [97], we first sort all classes into the alphabetical order according to their class name. The first 128 classes and the last 128 classes are considered as in-class and novel categories, respectively. The in-class categories are further divided into 50-50 splits to create training and test sets.

Caltech-UCSD Birds-200. The Caltech-UCSD Birds (CUB-200) is a fine-grained bird classification dataset. It contains 200 distinct bird categories and 6033 images in total. Similar to the protocol used before, the first 100 classes in the alphabetical order are picked as in-class categories and the last 100 classes in the alphabetical order are considered as the novel classes. The in-class

Table 6.1: Network architecture for novelty detection network.

Input
Conv, 7×7, 16, BatchNorm, LkyReLU
MaxPool, 2×2, stride 2
Conv, 7×7, 32, BatchNorm, LkyReLU
MaxPool, 2×2, stride 2
Conv, 7×7, 64, BatchNorm, LkyReLU
Fully Connected, 1280×128, BatchNorm, LkyReLU
Fully Connected, 128×2, SoftMax

categories are further divided into 50-50 splits to create training and test sets. As before, we make sure that both novel and in-class categories have equal number of images.

Stanford Dogs. This is another fine-grained classification dataset, containing 120 distinct dog breeds and a total of 20580 images. After sorting the dog breed classes in the alphabetical order, we pick the first and the last 60 breed categories as in-class and novel class, respectively. The in-class categories are further divided into 50-50 splits to create training and test sets. The number of images are the same for both in-class and novel classes during testing.

FounderType-200. The FounderType-200 dataset contains 200 different font types corresponding to the Chinese characters. Each font type category contains 6763 images. Similar to the other datasets, the first 100 font types are used as in-class categories and the last 100 font types are used as the novel class categories. We keep 50% of the image samples per category as the training set and the remaining 50% are used for testing. The number of images are the same for both in-class and novel classes during testing.

6.3.2 Training details

We first separately fine-tune the global and local inference networks with given in-class data. The global inference network is fine-tuned for 20000 iterations using SGD with 0.9 momentum, batch size of 64, initial learning rate of 0.001 which is decreased by a factor of 10 every 5000 iterations. The local inference network is fine-tuned till 40000 iterations using SGD with 0.9 momentum, batch size of 128, initial learning rate of 0.01 which is decreased by a factor of 10 every 10000 iterations. Both these fine-tuned networks then used to train the novelty detection network as described in this section. The novelty detection network is trained using SGD for 30000 iterations with 0.9 momentum, batch size of 32 and initial learning rate of 0.01 which is decreased by a factor of 10 every 10000 iterations. Images were resized to 256×256 pixels and a randomly cropped of size 224×224 pixels. Due to lack of enough images for each category we utilize data augmentation to increase the number of images for training. Specifically, for Caltech-256 and Dog-120 we use random flip, for CUB-200 we use random flip and horizontal-vertical translation of 10 pixels. Since for FounderType-200 dataset, each class contains approximately 7000 images and hence, no augmentation was used.

6.3.3 Network architecture

For global inference network we use AlexNet and VGG16 architecture as previously used in the literature [97]. For local inference network we used BagNet-33 architecture [8]. The BagNet architecture details can be found in [8]. It modifies the ResNet-50 architecture by replacing most 3×3 convolutions

Table 6.2: Novelty detection performance measured using the Area Under the receiver operating characteristic Curve evaluation metric (AUC). The best performing method for each dataset is shown in bold. The second best method is shown in italics. Here, symbol † indicate that reference dataset was used during training for that method.

Method	Caltech		CUB		Stanford Dogs		FounderType		Overall Performance
	VGG16	AlexNet	VGG16	AlexNet	VGG16	AlexNet	VGG16	AlexNet	
Fine-tune	0.827	0.785	0.931	0.909	0.766	0.702	0.841	0.650	0.801
K-extremes [117]	0.546	0.521	0.520	0.514	0.610	0.592	0.557	0.512	0.546
OC-SVM [114]	0.576	0.561	0.554	0.532	0.542	0.520	0.627	0.612	0.567
KNFST [7]	0.743	0.688	0.891	0.748	0.633	0.602	0.870	0.678	0.732
Local KNFST [6]	0.712	0.628	0.820	0.690	0.626	0.600	0.673	0.633	0.673
OpenMax [5]	0.831	0.787	0.935	0.915	0.776	0.711	0.852	0.667	0.809
Fine-tune† [97]	0.848	0.788	0.921	0.899	0.780	0.692	0.754	0.723	0.800
DTMND† [97]	*0.869*	0.807	0.958	0.947	0.825	0.748	*0.893*	0.741	0.848
Proposed	0.859	*0.826*	*0.972*	*0.952*	*0.827*	*0.751*	0.876	*0.798*	*0.857*
Proposed†	**0.870**	**0.847**	**0.979**	**0.965**	**0.873**	**0.812**	**0.898**	**0.801**	**0.879**

with 1×1 convolutions and changing the stride values to achieve independent patch-wise processing of input image. The network architecture used for novelty detection network is shown in Table 6.1. Here, LkyReLU denotes LeakyReLU activation with 0.2 negative slope and after every convolutional layer we apply dropout of with 0.2 probability 0.2.

6.3.4 Quantitative analysis

6.3.4.1 Novelty detection performance

We evaluate the performance of our method and compare it with several recent novelty detection methods. Each method provides a score to quantify the novelty of a test image. The lower the score, the higher the probability of input being from a novel class and vice versa. Following the protocol proposed in [97], we compare all methods using AlexNet [53] and VGG16 [123] as the global inference network architectures. In our approach, BagNet-33 [8] is used as the local inference network. Below is the list of methods used for comparison:

• **Fine-tune:** In this baseline, the pre-trained DCNN models are fine-tuned on the in-class data samples. The scores from penultimate layer of the models are used to evaluate novelty detection performance.

• **OC-SVM:** One-class SVM [114] is trained on the fine-tuned features and the SVM scores are used to evaluate the novelty detection performance.

• **KNFST:** KNFST as proposed in [6]. It uses fine-tuned deep features to learn a subspace for in-class data. The distance from the subspace is used to evaluate the performance.

• **Local KNFST:** Local KNFST [6] is an extension of the previous baseline, where a local region of in-class data are used to compute the score for performance evaluation.

• **OpenMax:** OpenMax [5] uses penultimate layer scores of a fine-tuned DCNN and distance from class-wise mean vectors combined with extreme value modeling for performance evaluation.

• **K-extremes:** This baseline focuses on the penultimate activations where top 10% of the sorted activations are binarized to find extreme signatures, which are later used to compute the normalized scores for performance evaluation.

• **Fine-tune†:** This is another fine-tuning baseline proposed in [97]. Here, during fine-tuning DCNN on any given novelty detection dataset, a *reference dataset* is used to improve the quality of the features. During testing, the maximum score from the penultimate layer of a DCNN, extracted from the in-class categories (excluding the reference dataset) is used for performance evaluation.

• **DTMND:** Recently proposed novelty detection method, where a *reference dataset* is utilized in a novel training strategy to learn better model that can

respond negatively to the novel classes. Maximum activation from the penultimate layer of the model is used for evaluating the novelty detection performance.

The evaluation protocol proposed by [97] considered two more baselines, namely KNFST-*pre* and Local KNFST-*pre*. However, we excluded these from comparison here as they do not observe any improvement over the KNFST and Local KNFST baselines. More details regarding these baselines are provided in [97]. For the proposed method, we use addition of scores from the global inference and the novelty detection networks to evaluate the performance.

The performance of different methods are evaluated using the area under the receiver operating characteristic curve (AUC) metric. The results are reported in Table 6.2. As can be seen from this table, OC-SVM and K-extremes methods have the lowest performances. Local KNFST performs better than both OC-SVM and K-extremes for all four datasets. KNFST provides better performance compared to Local KNFST on average, and has consistently better performance on all datasets. On average Fine-tune and Fine-tune† have similar performances. However, their performances are inconsistent across datasets and network architectures. For the Caltech-256 dataset, Fine-tune† performs better than Fine-tune for both AlexNet and VGG16, while for CUB-200 the trend is reversed. For both the Stanford Dogs and the FounderType-200 datasets, Fine-tune† performs better when the VGG16 architecture is used and the reverse trend is observed when the AlexNet architecture is used. The performance obtained by Fine-tune† baseline shows that simple fine-tuning is

not an efficient way to utilize a reference dataset for novelty detection. OpenMax performs better than both Fine-tune and Fine-tune† baselines, resulting in 1% overall improvement. Except for the FounderType-200 dataset using the VGG16 architecture, OpenMax consistently performs better than OC-SVM, K-extremes, Local KNFST, KNFST, Fine-tune and Fine-tune† baselines. Out of all the baselines, DTMND yields the best performance. DTMND on average performs 3% better than the next best performing baseline and performs approximately 5% better than Fine-tune† on average. Even-though both of these baselines have access to a reference dataset, DTMND utilizes this additional data more efficiently, resulting in the better performance. The performance of DTMND is largely attributed to their approach for fine-tuning using the reference dataset.

In the absence of reference dataset, the best method in the literature DTMND would become similar to that of fine-tune baseline and the performance will drop by $\sim$5% to 0.80. Whereas the proposed approach without the reference dataset during training provides approximately 6% improvement over the DTMND without reference dataset. This is due to the fact that the performance gain for DTMND is mainly due to the fact that it uses an external reference dataset for training the network. When the reference dataset is utilized during the training of the proposed approach (described in Eq. 6.5), the proposed approach consistently performs better than DTMND for all datasets and network architectures. Overall, when the proposed approach is trained with the help of reference dataset it improves by $\sim$2% and provides $\sim$4% improvement over the DTMND. The performance improvement with

the proposed† method shows that our approach can be easily extended to a scenario where a reference dataset is available to further enhance the novelty detection performance. On the other hand, DTMND becomes sub-optimal for the cases where a reference dataset is not available. Especially in such cases the proposed approach is a better alternative for DCNN-based multi-class novelty detection compared to DTMND.

6.3.4.2 Ablation analysis

In this section, we provide an ablation analysis showing the significance of combining patch-level information with global in our approach. For ablation experiments, we consider all four novelty detection datasets and the corresponding protocol proposed in Sec. 6.3.1. For all experiments, VGG16 is used as the global inference network. The following ablation baselines are considered:

• **Global Only:** This baseline is similar to Fine-tune as described in Sec. 6.3.4.1. The in-class data samples are used to fine-tune the VGG16 network. The maximum activation score from the penultimate layer of VGG16 is used to evaluate the novelty detection performance.

• **Local Only:** Fine-tuning only the local inference network using the given in-class data. The maximum activation score from the penultimate layer of the local inference network is used to evaluate the novelty detection performance.

• **Global+Local:** Here, we perform a straight forward concatenation of information from the global and local inference networks. The novelty detection performance is evaluated based on the addition of scores from both networks.

Table 6.3: Ablation analysis using AUC. The best performing method is shown in bold.

Method	Caltech	CUB	Stanford Dogs	FounerType	Overall Performance
Global Only	0.827	0.931	0.766	0.841	0.841
Local Only	0.799	0.785	0.598	0.773	0.739
Global+Local	0.831	0.943	0.741	0.835	0.837
Proposed	**0.859**	**0.972**	**0.827**	**0.876**	**0.883**

• **Proposed:** This is the method proposed in this section, where instead of a straight-forward fusion we utilize novel training strategy proposed in Sec. 6.1, to train a novelty detector network, which can better identify the mismatch of local activity patterns for global feature of a given category.

The performance of all three ablation baselines are reported in Table. 6.3. The lowest performance is obtained by local only baseline. The local inference network processes image patches and classifies images based on the local image features. This leads to relatively poor classification of in-class samples, which in turn hurts the novelty detection performance. On the other hand, the global inference network processes the entire image with a cascade of convolutional, pooling and fully connected layers to get a feature encoding for the entire image. This helps the global only baseline perform better classification and generates high prediction scores for the in-class samples. However, the problem with the global only baseline is that it ends up providing high prediction scores for the novel class samples as well, hurting the novelty detection performance. In the proposed approach, the novelty detection network is trained using both local and global inference networks. The combined information and novel training strategy helps the trained novelty detection network to perform better in identifying novel classes. Specifically, the local

inference network provides patch-level activation information corresponding to the prediction provided by the global inference network. The novelty detection network identifies the mismatch between the patch-level activation patterns and global feature encoding to predict whether the input image belongs to either in-class or novel class. As a result, the proposed method performs approximately 14% and 4% better than the local and the global baselines, respectively. We also compare the performance of our method with a *naive* fusion baseline, i.e. Global+Local, where the information from global and local networks are directly concatenated and the performance evaluation is done using the added scores. From Table. 6.3, it can be observed that the proposed approach is able to perform better than the Global+Local baseline.

6.3.5 Qualitative analysis

6.3.5.1 Fine-tune baseline vs proposed method

To show the effectiveness of the proposed approach, we provide a qualitative comparison with the Fine-tune baseline (i.e. traditional DCNN) in Fig. 6.4. Specifically, we provide image examples, prediction from the global inference network, their corresponding local class-activation heat-maps and scores assigned by both baseline and the proposed method. The heat-maps are generated by normalizing the local feature encodings of the class predicted by the global inference network. The images presented here are from two novel classes, namely, 'Tambourine' and 'Treadmill', as shown in Fig. 6.4(b), Fig. 6.4(d), respectively. These images are wrongly identified by the baseline as in-class data, and assigned the category 'Backpack', and 'Ladder' with high

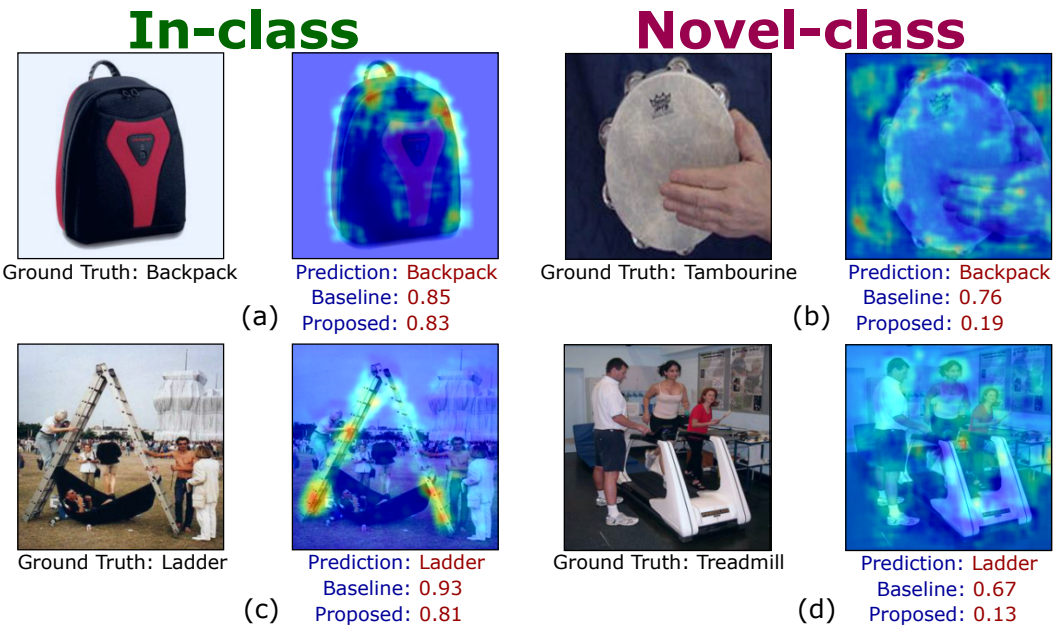


Figure 6.4: Image examples of in-class (a) & (c) and novel class (b) & (d) data with corresponding class activation heat-maps as predicted by local inference network and scores assigned using both baseline and proposed.

scores. Additionally, we show the images from the corresponding in-class categories 'Backpack' and 'Ladder' and their corresponding class activation heat-maps in Fig. 6.4(a) and Fig. 6.4(c), respectively. This figure shows the difference in class activation heat-maps for the case where the image samples are from in-class data and the case where the image samples are from novel classes. For example, in Fig. 6.4(a), the image sample is from a known class with category label 'Backpack' and the network is able to correctly identify it by assigning a high score. The patch-level class activation patterns shown in heat-map focuses on highly discriminative patch locations providing strong presence of the given class. On the other hand, in Fig. 6.4(b), the image sample is from a novel class, but the network wrongly identifies it as 'Backpack' with a high score. However, if we look at the class activation patterns, there are

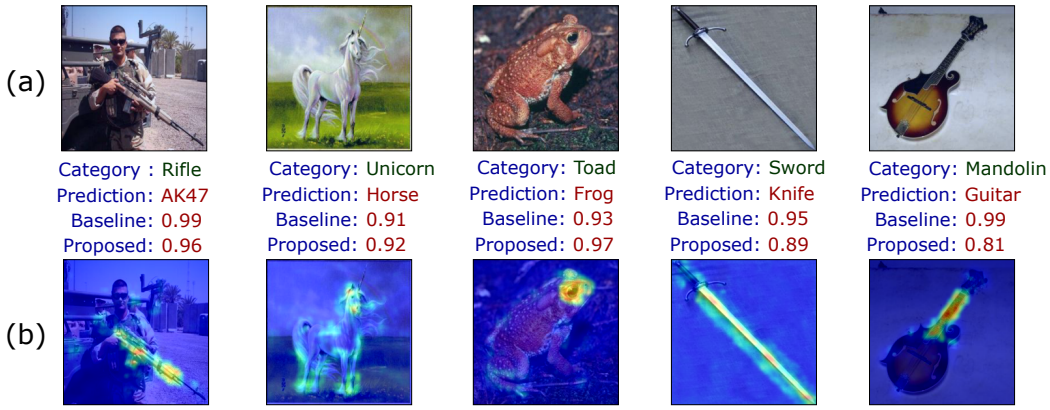


Figure 6.5: Examples of images from novel classes that are wrongly identified as in-class samples with high scores.

moderate to high activations all over the image, as opposed to in-class image in Fig. 6.4(a). The novelty detector of the proposed method is specifically trained to identify this mis-match in activation patterns and predicted label. This helps the proposed approach correctly predict a high score for the image sample of 'Backpack' and a low score for the image sample of a novel class, 'Tambourine'. Similar observations can be made for the other example provided for 'Ladder' in Fig. 6.4(c) and Fig. 6.4(d).

Though the proposed approach exhibits reasonable novelty detection performance, there are some cases where it fails to predict low scores when the samples are from novel classes. Some of these examples are illustrated in Fig. 6.6 with their corresponding class activation heat-maps and the predicted scores using the Fine-tune baseline (i.e. traditional DCNN) and the proposed method. The image sample from novel category 'Toad' is identified as in-class category 'Frog'. In this case, the novelty detector network fails to detect any mis-match between the local patch-wise activation patterns and the predicted label. Similarly, the novel categories 'Unicorn', 'Rifle' and 'Mandolin' are

Figure 6.6: Examples of images and their corresponding patch-level activity patterns, from in-class categories that are wrongly identified as novel class samples with low prediction scores.

identified as in-class categories 'Horse', 'AK47' and 'Guitar', respectively. For all of these examples presented here, the reason for failure can be due to very subtle differences between these novel categories with their respective mis-classified in-class categories.

Here, we provide examples of in-class samples that are miss-identified as novel with low prediction score. In some cases the image examples are miss-classified as in the case of Baseball, Billiards and Dice where network has high activation from wrong local-patches of the image. In other cases such as Telescope and Pisa-tower, the miss-identification happens due to the difficulty of image examples. E.g., in Pisa-tower image example the tower of Pisa is around cluster of other buildings and hence results in low prediction score. In the case of Telescope example the telescope is occluded by Person in the front and hence receives low activation across informative image patches, resulting in low prediction score for the image.

6.3.6 Conclusion

We proposed a novel DCNN-based multi-class novelty detection method, that is end-to-end trainable. Unlike recent methods, the proposed approach does not rely on the availability of a reference dataset and is flexible enough to work on both scenarios, when the reference dataset is available and when it is not. We discussed assumptions regarding patch-level activation patterns of DCNNs when the test image is from novel classes. Based on these assumptions, we proposed a novel training methodology which utilizes both global level predictions from the traditional DCNNs and a local inference network, which processes image at patch level. Furthermore, we show how the proposed approach can be extended when a reference dataset is accessible by regularizing the reference data penultimate activations. Experimental results, evaluated on four multi-class novelty detection datasets, show that the proposed method is able to identify novel class samples better compared to the other DCNN-based methods.

Chapter 7

Multi-class Novelty Detection under Distribution Shift

7.1 Motivation

Most of the existing works tackling the issue of multiple-class novelty detection, try to learn a decision boundary that encloses the known categories given in the dataset. However, while trying to enclose the known categories, these methods also enclose the style/domain of the dataset. As a result, samples from known categories but having different style/domain, will have increased risk of false detection as a novel category. For example, a novelty detection method trained on SVHN digits dataset will be correctly able to detect known categories from novel, only if the test data follows the same distribution as SVHN. But, if the test data is from a digits dataset like MNIST, due to the domain shift, it is highly likely that the novelty detector will not be able to distinguish between novel and known categories accurately. This problem is also illustrated in Fig. 7.1. Most of the earlier novelty detection methods work on the assumption that the test data would follow a similar distribution as the

training data.

We also provide a preliminary experimental analysis to show the effect of dataset distribution shift on the performance of novelty detection. For this experiment, we consider a novelty detector [105], referred to as Adversarially

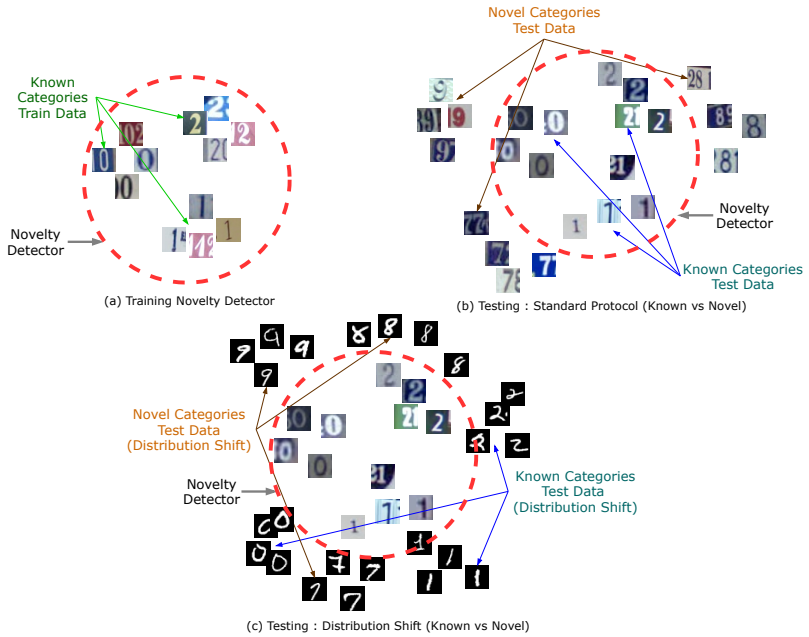
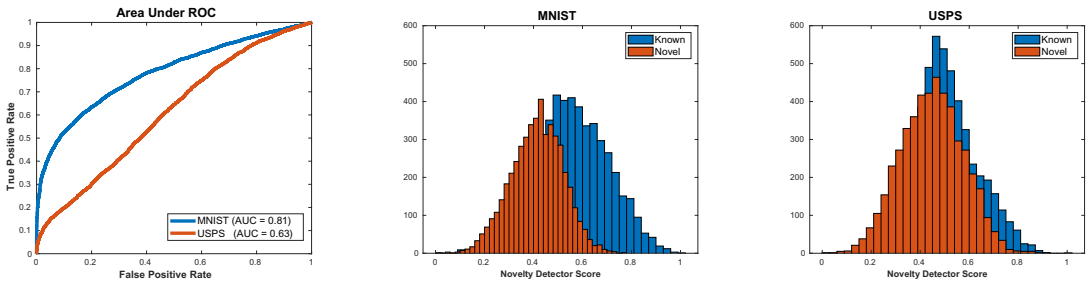


Figure 7.1: An overview of the proposed problem setting. (a) We have a training data with samples from multiple known categories. Here, we have used the SVHN dataset with digits 0, 1 and 2 as known categories. These data samples are used to learn a novelty detector to enclose the known categories. (b) In a standard novelty detection testing protocol, the test data follows the same distribution as the training data. As shown in the figure, typically the novelty detector is able to distinguish between known categories and novel categories. Here, digits 7, 8 and 9 sampled from the SVHN dataset are used as novel categories. As illustrated in the figure, the learned novelty detector is able to differentiate between known and novel digits from the SVHN dataset correctly. (c) This figure illustrates the scenario where the test data does not follow the distribution of the training dataset. When tested with known (0, 1, 2) and novel (7, 8, 9) digits from the MNIST dataset, due to the distribution shift, the learned novelty detector performs poorly. This problem arises due to the fact that while training any novelty detector to enclose the known categories of a particular dataset, it also encloses the style/domain of that dataset. This creates a problem as shown in this figure, where the data from known categories, which follow a different distribution will have high risk of being detected as novel category.



(a) (b) (c)

Figure 7.2: (a) Area under the ROC plot when a novelty detector is evaluated on the MNIST and USPS datasets. (b) Histogram of scores corresponding to the MNIST dataset. (c) Histogram of scores corresponding to the USPS dataset.

learned One-Class Classifier (ALOCC). The ALOCC method is trained on the MNIST dataset. For training, we consider digits 0 to 4 as known categories and the remaining digits as novel categories. Fig. 7.2(a) shows the ROC curve illustrating the performance of the novelty detector when evaluated on the MNIST data (Blue curve). The novelty detector achieves area under the curve of 0.81. In order to simulate the data distribution shift, we evaluate the novelty detector on the USPS dataset, again considering 0 to 4 digits as known categories and the remaining digits as novel categories. As we can see from Fig. 7.2(a), the performance on the USPS dataset (red curve) drops by $\sim$20% compared to the MNIST dataset. Also, by looking at the histogram of score predictions in Fig. 7.2(b) and Fig. 7.2(c), it is clear that compared to MNIST, USPS scores for both known and novel categories on average are shifted towards the left. This shows that the novelty detector trained on MNIST has high risk of detecting USPS known categories as novel. This is due to the shift in the distribution between MNIST and USPS datasets.

Hence, in this section, we consider the problem of multiple-class novelty detection under dataset distribution shift. Since no prior work has been done

for this specific problem, we first describe the problem statement in detail and provide trivial baselines for this task based on novelty detection and domain adaptation approaches. Furthermore, we propose a novelty detection method that can address the data distribution shift problem and help improve over the trivial baselines. Moreover, we discuss the differences between the closely related problem setting such as open-set domain adaptation [87] and also provide experimental analysis to show that their performance is sub-optimal in the problem setting considered in this paper.

7.2 Robust novelty detection under distribution shift

In this section, we first formulate the problem and then discuss some baseline methods. Finally, we present the proposed method in detail.

7.2.1 Problem setting

Typically, a novelty detection model is developed using a training dataset having multiple categories which we refer to as the source dataset. This trained model is then tested in the real-world where the goal is to detect any test input samples belonging to novel categories. However, these models have high risk of detecting any test samples belonging to known categories as unknown, when the test samples are from a different distribution than that of the training dataset. The goal of the proposed problem setting is to generalize the novelty detection models on a dataset having different distribution, which we refer to as the target dataset. The terminology of referring labeled dataset as source and unlabeled dataset as target is borrowed from the domain adaptation

literature. Formally, in the proposed problem setting, we have access to the source dataset, $\mathcal{D}_s = \{X_{si}\}_{i=1}^{N_s}$ and their corresponding label set $\mathcal{Y}_s = \{y_{si}\}_{i=1}^{N_s}$. There are in total C categories and each y_{si} takes a value from the label set $\{1, 2, ..., C\}$. Similarly, we have access to the target dataset, $\mathcal{D}_t^k = \{X_{ti}\}_{i=1}^{N_t}$, having different distribution than the source dataset. Both source ($\mathcal{D}_s$) and target ($\mathcal{D}_t^k$) datasets share the same C categories. However, for $\mathcal{D}_t^k$ we do not have access to the corresponding labels. Here, the superscript k denotes that the dataset contains only the known categories, i.e., all data samples in the $\mathcal{D}_t^k$ belong to one of the categories from the label set $\{1, 2, ..., C\}$. During training, the goal is to learn a novelty detector that generalizes well on the target dataset with the help of the information available in the source dataset, i.e., $\mathcal{D}_s$ and $\mathcal{Y}_s$. The learned novelty detector is evaluated using a test set from the target dataset ($\mathcal{D}_t^{k:test}$) having known categories and a target set containing data from unknown categories ($\mathcal{D}_t^u$). Here, superscript u denotes that the dataset contains only novel categories. Note that data from $\mathcal{D}_t^u$ is not utilized during training but only used while evaluating the novelty detection performance on the target set.

7.2.2 Simple approaches

As shown by preliminary experiment in Sec. 7.1 the dataset distribution shift is one of the unexplored problems in novelty detection. Following the problem setting and notations described in previous section, in this section, we explore some potential solutions for tackling this problem. Since there are no prior works available in the literature on this problem, we develop a few baselines

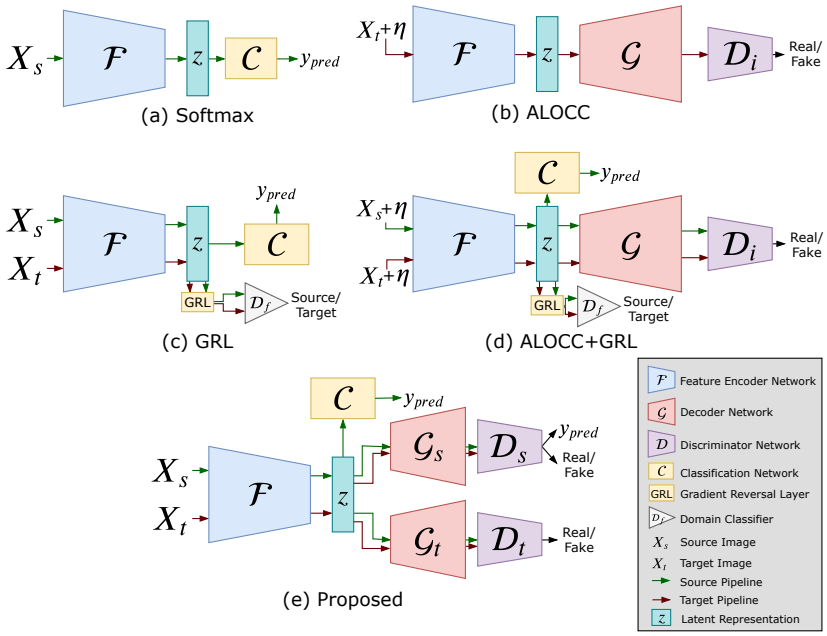


Figure 7.3: Illustration of multiple potential solutions to address the distribution shift problem for novelty detection. (a) Softmax: Simplest approach which utilizes the labeled source data to train a classification network. The maximum softmax probability can be used as the novelty score. (b) ALOCC: Another approach which directly utilizes the unlabeled target data to train an off-the-shelf novelty detector. We utilize, a novelty detection algorithm proposed in [105]. Here, η denotes the Gaussian noise added to the input image. (c) GRL: Uses labeled source and unlabeled target data to learn a domain invariant feature space using a gradient reversal layer [30]. The maximum softmax prediction probability can be used as the novelty score. (d) ALOCC+GRL: A combination of both novelty detector [105] and domain invariant feature learning [30] in an ad-hoc manner. (e) Proposed method: A shared feature space is learned through cross-domain mappings. The corss-domain mappings helps to learn a better feature space which is especially useful for novelty detection.

by considering similar works from the literature. The block diagrams of these methods are illustrated in Fig. 7.3(a)-(d). In what follows, we describe these baseline approaches in detail.

Softmax. The most simple baseline would be to utilize the labeled source data to train a feature extractor and classifier network to perform multi-class classification. However, classification networks are prone to novel classes even

in the source domain, hence would not translate well for the target domain novelty detection.

ALOCC. Another approach would be to disregard the source domain information and only use the target domain unlabeled data to train any off-the-shelf novelty detector algorithm. For this baseline, we utilize ALOCC method for novelty detection proposed in [105]. Specifically, ALOCC trains an auto-encoder which aims to reconstruct a clean image from the input image using Gaussian noise. This auto-encoder network is trained in generative adversarial framework and the score from the discriminator of the reconstructed image is used for novelty detection. The dataset will have multiple categories, however ALOCC remains agnostic to that by considering multiple categories as one.

GRL. Gradient reversal layer [30] has been widely used to reduce the domain gap between two datasets having different distributions for the classification task. GRL baseline can be considered as an extension to the Softmax baseline such that the domain gap issue between source and target is addressed by the gradient reversal layer.

ALOCC+GRL. This is the final baseline which combines the gradient reversal training to reduce the domain gap between source and target, together with the novelty detection training specified in the ALOCC. This ad-hoc combination provides a strong baseline for the proposed setting, since GRL is able to take care of the domain gap and with the help of domain invariant feature space, the ALOCC is able to learn a more general novelty detector which is likely to perform better on the target domain.

7.2.3 Proposed method

ALOCC+GRL is the most related method out of all the methods described above. Also, it is able to exploit both novelty detection training and domain adversarial loss to learn a domain invariant feature space. This should help the novelty detector mitigate the effects of distribution shift and perform reasonably well on the target domain. However, such method is an ad-hoc combination of the domain adaptation and novelty detection algorithms. To get the best out of the information available in the proposed problem setting, we need a unified approach where novelty detection training inherently mitigates the distribution shift. Fig. 7.3(e) gives an overview of the proposed approach, where the cross-domain decoders trained for novelty detection task guides the shared feature extractor to learn a common feature space. As opposed to the method with ad-hoc combination, the proposed way of learning can benefit from the unified training strategy, since the novelty detection task guides the feature space learning. Here, we discuss the training methodology used for proposed approach.

Let's consider images X_s and X_t sampled from the source and target domain, respectively. The feature encoder network ($\mathcal{F}$), takes these samples and generates latent representations z_s and z_t. Since, for the source domain, we have access to the class labels, the classifier ($\mathcal{C}$) is trained to classify latent representations of source domain in to respective categories. As discussed earlier, the feature extractor network $\mathcal{F}$ is learned with the help of two generator networks $\mathcal{G}_s$ and $\mathcal{G}_t$ for source and target domain, respectively.

For the source domain discriminator D_s, a conditional GAN [82] based

approach is used. This specifically helps the generator networks when datasets contain multiple categories. Following the conditional GAN formulation proposed by [82], the discriminator network D_s has two parts. The first part referred to as, D_s^b, identifies whether the samples generated by $\mathcal{G}_s$ are real or fake by a binary classification. On the other hand, the second part referred to as, D_s^a, classifies the generated images into one of the known categories. $\mathcal{G}_s$ takes in the latent representations z_s and z_t to generate images $\hat{X}_{s2s}$ and $\hat{X}_{t2s}$, respectively. This process can be described as follows,

$$z_s = \mathcal{F}(X_s), \;\; z_t = \mathcal{F}(X_s)$$

$$\hat{X}_{t2s} = \mathcal{G}_s(z_t), \;\; \hat{X}_{s2s} = \mathcal{G}_s(z_s). \tag{7.1}$$

For the target domain discriminator D_t, a binary classifier based on the cross entropy loss is used. The generator network $\mathcal{G}_t$ generates the image samples from the source and the target domain, using latent representations z_s and z_t, respectively. This process can be described as follows,

$$\hat{X}_{s2t} \;=\; \mathcal{G}_t(z_s), \quad \hat{X}_{t2t} \;=\; \mathcal{G}_t(z_t). \tag{7.2}$$

The classifier loss function can be defined as follows

$$\mathcal{L}_{ce} = \; \mathbb{E}_{\{X,y\}\sim\{\mathcal{D}_s,\mathcal{Y}_s\}}[\ell_{ce}(\mathcal{C}(\mathcal{F}(X)), y)], \tag{7.3}$$

where, $\mathcal{L}_{ce}$ is the overall classification loss computed on the labeled source data and ℓ_{ce} is the categorical cross entropy loss. Considering $\hat{y} = \mathcal{C}(z_s)$ as the predicted probability vector, ℓ_{ce} can be expressed as follows

$$\ell_{ce}(\hat{y}, y) \;=\; -\sum_{j=1}^{C} y_j \;\log[\hat{y}_j]. \tag{7.4}$$

To train the source discriminator in the conditional GAN framework, we need

to perform real/fake classification and categorical classification, which can be expressed as

$$\mathcal{L}_{cGAN}^{D_s} = \mathbb{E}_{X\sim\mathcal{D}_s}[\log(1 - D_s^b(X))] + \mathbb{E}_{X\sim\mathcal{D}_s}[\log(D_s^b(\hat{X}_{t2s}))]$$

$$+ \mathbb{E}_{X\sim\mathcal{D}_t^k}[\log(D_s^b(\hat{X}_{s2s}))] + \mathbb{E}_{X\sim\mathcal{D}_s, y\sim\mathcal{Y}_s}[\ell_{ce}(D_s^a(\hat{X}_{s2s}), y)], \quad (7.5)$$

where, the first term in the equation trains the discriminator D_s^b to identify data sampled from the source dataset $\mathcal{D}_s$ as real images. The second and third term train the discriminator to identify images generated by $\mathbb{G}_s$, i.e., $\hat{X}_{t2s}$ and $\hat{X}_{s2s}$, as fake. The fourth term is a classification loss similar to Eq. 7.3, where the generated images $\hat{X}_{s2s}$ are classified in to the category corresponding to the source input images using D_s^a.

After the discriminator update, the source generator is trained to generate images such that the discriminator network is fooled into identifying the generated images, $\hat{X}_{s2s}$ and $\hat{X}_{t2s}$ as real source images. To further improve the image generation quality, we add L1 reconstruction loss, denoted as ℓ_r, on the generated source images, $\hat{X}_{s2s}$. The loss functions described above can be mathematically formulated as

$$\mathcal{L}_{cGAN}^{\mathcal{G}_s} = \mathbb{E}_{X\sim\mathcal{D}_s}[\log(1 - D_s^b(\mathcal{G}_s(X)))] + \mathbb{E}_{X\sim\mathcal{D}_t^k}[\log(1 - D_s^b(\mathcal{G}_s(X)))], \quad (7.6)$$

$$\mathcal{L}_{rs}^{\mathcal{G}_s} = \mathbb{E}_{X\sim\mathcal{D}_s}[\ell_r(\hat{X}_{s2s}, X)], \quad (7.7)$$

where

$$\ell_r(\hat{X}, X) = \|X - \hat{X}\|_1. \quad (7.8)$$

Similar to the source domain discriminator and generator, we apply the same

GAN losses for the target domain discriminator D_t, and generator $\mathcal{G}_t$. Since, the target domain labels are not available, a traditional GAN formulation is used [33], instead of the conditional GAN formulation [82] used for source domain. Additionally, similar to the source domain, we add L1 reconstruction loss on the generated target images, $\hat{X}_{t2t}$, to further improve the image generation quality in the target domain. These losses can be written as follows

$$\mathcal{L}_{GAN}^{D_t} = \mathbb{E}_{X\sim\mathcal{D}_t}[\log(1 - D_t(X))] + \mathbb{E}_{X\sim\mathcal{D}_s}[\log(D_t(\hat{X}_{s2t}))] + \mathbb{E}_{X\sim\mathcal{D}_t^k}[\log(D_t(\hat{X}_{t2t}))], \tag{7.9}$$

$$\mathcal{L}_{GAN}^{\mathcal{G}_t} = \mathbb{E}_{X\sim\mathcal{D}_t^k}[\log(1 - D_t(\mathcal{G}_t(X)))] + \mathbb{E}_{X\sim\mathcal{D}_s}[\log(1 - D_t(\mathcal{G}_t(X)))], \tag{7.10}$$

$$\mathcal{L}_{rt}^{\mathcal{G}_t} = \mathbb{E}_{X\sim\mathcal{D}_t^k}[\ell_r(\hat{X}_{t2t}, X)]. \tag{7.11}$$

Finally, the loss function for the feature encoder network consists of both the classification loss on the source and the adaptation loss from the conditional GAN module. The final loss for the network $\mathcal{F}$ can be expressed as

$$\mathcal{L}_{total}^{\mathcal{F}} = \mathcal{L}_{ce} + \lambda_1 \mathcal{L}_{cGAN}^{\mathcal{G}_s} + \lambda_2 \mathcal{L}_{GAN}^{\mathcal{G}_t}, \tag{7.12}$$

where λ_1 and λ_2 are parameters. The loss functions defined above, $\mathcal{L}_{cGAN}^{\mathcal{G}_s}$, $\mathcal{L}_{cGAN}^{D_s}$, $\mathcal{L}_{GAN}^{\mathcal{G}_t}$, $\mathcal{L}_{GAN}^{D_t}$, $\mathcal{L}_{ce}^{\mathcal{C}}$, $\mathcal{L}_{total}^{\mathcal{F}}$, $\mathcal{L}_{rt}^{\mathcal{G}_t}$ and $\mathcal{L}_{rs}^{\mathcal{G}_s}$, are minimized iteratively to update the parameters of their respective networks. The overall training procedure for the proposed method is summarized in Algorithm 1.

7.3 Experiments and results

For experiments, we consider all the baseline methods discussed in Sec. 7.2.2 and the proposed method described in Sec. 7.2.3. We use SVHN [81], MNIST

Figure 7.4: Sample images from the datasets used for conducting experiments. (a) Digits (b) Office-31.

[59] and USPS [46] digit recognition datasets, as well as the Office-31 [106] object recognition datasets to conduct experiments (see Fig. 7.4). We evaluate the performance of different methods using the Area Under the ROC (AUROC) Curve metric, which is the most commonly used evaluation metric for novelty

---

Algorithm 1 Pseudocode for training proposed method

Require: Network models $\mathcal{F}, \mathcal{C}, \mathcal{G}_s, \mathcal{D}_s, \mathcal{G}_t, \mathcal{D}_s$
Require: Initial parameters $\Theta_f, \Theta_c, \Theta_{g_s}, \Theta_{d_s}, \Theta_{g_t}, \Theta_{d_t}$
Require: Source data, $\mathcal{D}_s$, $\mathcal{Y}_s$ Target data , $\mathcal{D}_t^k$
Require: Hyper-parameters : $N, lr, \lambda_1, \lambda_2$
while not done **do**
  for each batch with size N **do**
    for i = 1 to N **do**
      Feed-forward using Eq. (7.1) – Eq. (7.2)
    end for
    Calculate Losses based on Eq. (7.3) – Eq.(7.12)
    Update Θ_{d_s}, $\Theta_{d_s} \leftarrow \Theta_{d_s} - lr * \nabla_{\Theta_{d_s}} \mathcal{L}_{cGAN}^{D_s}$
    Update Θ_{d_t}, $\Theta_{d_t} \leftarrow \Theta_{d_t} - lr * \nabla_{\Theta_{d_t}} \mathcal{L}_{GAN}^{D_t}$
    Update Θ_{g_s}, $\Theta_{g_s} \leftarrow \Theta_{g_s} - lr * \nabla_{\Theta_{g_s}} \mathcal{L}_{cGAN}^{G_s}$
    Update Θ_f, $\Theta_f \leftarrow \Theta_f - lr * \nabla_{\Theta_f} \mathcal{L}_{total}^{\mathcal{F}}$
    Update Θ_c, $\Theta_c \leftarrow \Theta_c - lr * \nabla_{\Theta_c} \mathcal{L}_{ce}$
    Update Θ_{g_t}, $\Theta_{g_t} \leftarrow \Theta_{g_t} - lr * \nabla_{\Theta_{g_t}} \mathcal{L}_{rt}^{G_t}$
    Update Θ_{g_s}, $\Theta_{g_s} \leftarrow \Theta_{g_s} - lr * \nabla_{\Theta_{g_s}} \mathcal{L}_{rs}^{G_s}$
  end for
end while
Output: Learned parameters $\hat{\Theta}_f$,$\hat{\Theta}_s$,$\hat{\Theta}_{g_s}$,$\hat{\Theta}_{d_s}$,$\hat{\Theta}_{d_t}$,$\hat{\Theta}_{g_t}$

---

detection. Each datasets are divided into known and novel categories for novelty detection. Details regarding the splits are described in the following sections. The novel categories are not utilized during training and only used during inference. The following methods are comapred.

• **Softmax baseline:** In this baseline, only the feature extractor network $\mathcal{F}$ and the classification network $\mathcal{C}$ are trained on the labeled source dataset using the cross entropy loss. This is the simplest baseline and follows the traditional CNN training for recognition. Maximum softmax probability score is used for novelty detection.

• **ALOCC:** ALOCC is a method proposed in [105], which utilizes a feature extractor network $\mathcal{F}$ and a decoder network $\mathcal{G}$ supervised in a generative adversarial framework with the help of a discriminator network D_i. The training is done directly on the unlabeled target data. The input is injected with a Gaussian noise η and networks $\mathcal{F}$ and $\mathcal{G}$ are forced to reconstruct a clean image. The network parameters are learned by optimizing a combination of GAN and reconstruction losses. The discriminator score of the reconstructed input $D(\mathcal{G}(\mathcal{F}(X+\eta)))$ is used for novelty detection.

• **GRL:** Gradient reversal baseline extends the softmax baseline by improving the feature space to be domain invariant. This makes the maximum softmax probability much more reliable for the novelty detection task on the target domain. For GRL, feature extractor $\mathcal{F}$ and classifier network $\mathcal{C}$ are trained using the cross entropy loss and domain classifier D_f is employed with a gradient reversal layer [30] to enforce the feature space to be domain invariant. Here, the method utilizes both labeled source data and unlabeled target data

for training the network parameters.

• **ALOCC+GRL:** ALOCC+GRL combines the two method described above in an ad-hoc fashion. The ALOCC training is done as described above, which involves reconstructing a clean image when the input to the network is injected with Gaussian noise. For this baseline we add noise to both source and target data. The feature extractor network $\mathcal{F}$ is also trained to perform classification of labeled source data through classification network $\mathcal{C}$. Additionally, the feature space of network $\mathcal{F}$ is enforced to be domain invariant through domain classifier D_f and gradient reversal layer. Combination of scores from ALOCC and maximum softmax probability is used to perform novelty detection. The training utilizes both labeled source and unlabeled target data.

• **Proposed method:** The proposed method is used as described in Sec. 7.2.3. We use addition of maximum softmax probability scores and loss from target generator (i.e. discriminator score of generated image and reconstruction loss) for novelty detection.

In all experiments, we use Adam optimizer [51] with the learning rate (η) of 0.0001 and batch size (N) of 64. The hyper-parameter λ_1 and λ_2 are both set equal to 0.03. The parameters are chosen using validation performance from the source domain data. Details regarding the network architectures used for $\mathcal{F}$, $\mathcal{C}$, $\mathcal{G}_s$, $\mathcal{G}_t$, D_s and D_t are provided in supplementary material.

7.3.0.1 Digits: SVHN, USPS, MNIST

In the first set of experiments, SVHN, USPS and MNIST digit datasets are used to create four different scenarios, SVHN→MNIST, SVHN→USPS, USPS→MNIST

Table 7.1: Performance on the digits datasets - SVHN, MNIST and USPS evaluated using area under the roc metric. (S), (T) and (ST) respectively denote only labeled source data, only unlabeled target data and both labeled source-unlabeled target data used for training.

Method	SVHN→MNIST	MNIST→USPS	USPS→MNIST	SVHN→USPS	Average Performance
Softmax (S)	0.642	0.602	0.651	0.587	0.620
ALOCC (T)	0.702	0.633	0.702	0.633	0.667
GRL (ST)	0.718	0.863	0.859	0.667	0.776
ALOCC+GRL (ST)	0.851	0.903	0.895	0.845	0.873
Proposed (ST)	**0.919**	**0.945**	**0.928**	**0.895**	**0.921**

and MNIST→USPS. First five digits, digits 0 to 4, are used as known categories and the remaining digits, digit 5 to 9, are considered as novel categories. Only the known categories are used during training and novel categories are used only for evaluating the methods. For the problem setting proposed in this section, we utilize training split provided by the respective datasets to train the models and test split are used for evaluating the performance. All images in SVHN, MNIST and USPS are resized to 32×32. The feature extractor used in this section is inspired from the LeNet architecture [58] (details are provided in supplementary material).

The performance of each method is reported in the Table. 7.1. The softmax baseline performs worst out of all the methods. This is expected as softmax baseline is trained on only labeled source dataset. Also, it is not specifically trained for the novelty detection task. ALOCC performs better than softmax as it is trained on the target dataset and is specifically designed for the task of novelty detection. GRL baseline learns a domain invariant feature encoder, and hence is able to produce reasonable softmax probabilities on the target dataset. ALOCC+GRL combines the ideas from domain adversarial training and novelty detection training. Specifically, ALOCC learns a good model for novelty detection task and GRL helps the feature extractor of the ALOCC

model to learn domain invariant feature. Additional training with classification loss on the labeled source data helps the ALOCC+GRL to better utilize multi-class structure of the dataset, making it the best performing method among the baselines. All of the above methods are simple extensions or ad-hoc combinations of the work available in the literature. Whereas, the proposed approach tackles the distribution shift issue along with novelty detection training in a single model. This helps the proposed approach perform better than the ad-hoc solutions, performing $\sim 5\%$ better than ALOCC+GRL.

7.3.0.2 Office31 : Amazon, Webcam, DSLR

Finally, we evaluate the proposed method on the Office31 benchmark [106]. The Office31 benchmark has a total 31 object categories and three different domains. Image samples for the dataset are acquired in three different domains, i.e. Amazon (A), Webcam (W) and DSLR (D). First 10 categories from all three domains are considered as known. Categories from $11, 12, ..., 30$ are considered as novel categories for all domains. For all the methods compared, AlexNet [54] is used as the base feature extractor. During training we freeze all the convolutional layers of AlexNet and only fine tune the fully-connected layers. For training the generator networks $\mathcal{G}_s$ and $\mathcal{G}_t$ we resize the images to 32×32 and the discriminator architectures are used accordingly (more details in supplementary material). Three domains of the dataset form in total 6 pairs of source$\rightarrow$target combinations. For each source$\rightarrow$target combination, we report AUROC performance.

The peformance of each method is reported in Table 7.2. Overall the trend

Table 7.2: AUC performance of different methods on the Office31 [106] datasset.

Methods	**A→D**	**A→W**	**W→A**	**W→D**	**D→A**	**D→W**	**Average**
Softmax	0.719	0.835	0.655	0.862	0.606	0.842	0.737
ALOCC	0.776	0.725	0.608	0.983	0.570	0.884	0.758
GRL	0.766	0.730	0.624	**0.988**	0.572	0.890	0.762
ALOCC+GRL	0.783	0.759	0.640	0.987	0.576	0.898	0.774
Proposed	**0.877**	**0.863**	**0.824**	0.938	**0.807**	**0.940**	**0.877**

of performance improvements are similar to the digits experiment. Among all the methods, softmax baseline achieves the lowest performance. ALOCC improves by $\sim$ 2% over the softmax baseline, while GRL is able to improve $\sim$ 1% over ALOCC. Utilizing gradient reversal along with ALOCC training further improves the performance by $\sim$ 1%. The proposed approach on average performs better than the other approaches. Specifically, the proposed approach on average provides $\sim$ 9% improvement over the next best baseline of ALOCC+GRL.

7.3.1 Conclusion

We considered the problem of novelty detection under dataset distribution shift and showed the challenges it poses with experiments. To the best of our knowledge, this is the first work to address such problem for novelty detection. We also discussed the differences between the proposed problem setting and some of the related problems like open-set domain adaptation. We also developed a few trivial baseline methods based on the related works available in the literature by combining the techniques from novelty detection and domain adaptation. Finally, we proposed an approach to tackle the distribution shift by learning a shared feature space that can generalize better

in comparison with the baseline methods.

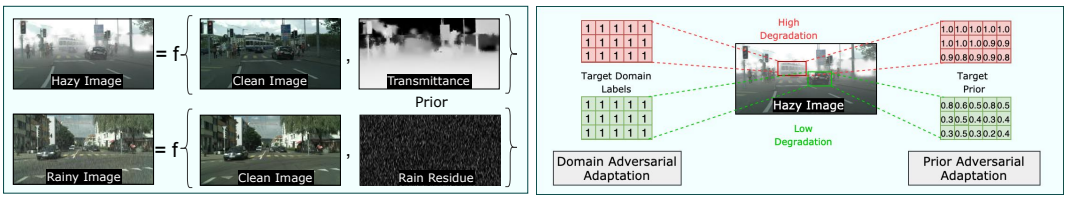
Chapter 8

Prior-based Domain Adaptive Object Detection

8.1 Motivation

Recent methods proposed to perform detection under adverse weather conditions, consider that the images captured under adverse conditions (target images) suffer from a distribution shift [16, 34] as compared to the images on which the detectors are trained (source images). It is assumed that the source images are fully annotated while the target images (with weather-based degradations) are not annotated. They propose different techniques to align the target features with the source features, while training on the source images. These methods are inherently limited in their approach since they employ only the principles of domain adaptation and neglect additional information that is readily available in the case of weather-based degradations.

We consider the following observations about weather-based degradations which have been ignored in the earlier work. ***(i)*** Images captured under weather conditions (such as haze and rain) can be mathematically modeled



(a) (b)

Figure 8.1: (a) Weather conditions such as rain and haze can be mathematically modeled as function of clean image and the weather-specific prior. We use this weather-specific prior to define a novel prior-adversarial loss for adapting detectors to adverse weather. (b) Existing domain adaptation approaches use constant target domain label for the entire image irrespective of the amount of degradation. Our method uses spatially-varying priors that are directly correlated to the amount of degradations.

(see Fig. 8.1(a)). For example, a hazy image is modeled by a superposition of a clean image (attenuated by transmission map) and atmospheric light [27, 38]. Similarly, a rainy image is modeled as a superposition of a clean image and rain residue [62, 139, 141] (see Fig. 8.1(a)). In other words, a weather-affected image contains weather specific information (which we refer to as prior) - transmission map in the case of hazy images and rain residue in the case of rainy images. These weather-specific information/priors cause degradations in the feature space resulting in poor detection performance. Hence, in order to reduce the degradations in the features, it is crucial to make the features weather-invariant by eliminating the weather-specific priors from the features. ***(ii)*** Further, it is important to note that the weather-based degradations are spatially varying and, hence do not affect the features equally at all spatial locations. Since, existing domain-adaptive detection approaches [16, 107, 121] label all the locations entirely either as target, they assume that the entire image has undergone constant degradation at all spatial locations (see Fig. 8.1(b)). This can potentially lead to incorrect alignment, especially in the

regions of images where the degradations are minimal.

Motivated by these observations, we define a novel prior-adversarial loss that uses additional knowledge about the target domain (weather-affected images) for aligning the source and target features. Specifically, the proposed loss is used to train a prior estimation network to predict weather-specific prior from the features in the main branch, while simultaneously minimizing the weather-specific information present in the features. This results in weather-invariant features in the main branch, hence, mitigating the effects of weather. Additionally, the proposed use of prior information in the loss function results in spatially varying loss that is directly correlated to the amount of degradation (as shown in Fig. 8.1(b)). Hence, the use of prior can help avoid incorrect alignment.

8.2 Proposed method

We assume that labeled clean data ($\{x_i^s, y_i^s\}_{i=1}^{n_s}$) from the source domain ($\mathcal{S}$) and unlabeled weather-affected data from the target domain ($\mathcal{T}$) are available. Here, y_i^s refers to all bounding box annotations and respective category label for the corresponding clean image x_i^s, x_i^t refers to the weather-affected image, n_s is the total number of samples in the source domain ($\mathcal{S}$) and n_t is the total number of samples in the target domain ($\mathcal{T}$). Our goal is to utilize the available information in both source and target domains to learn a network that lessens the effect of weather-based conditions on the detector. The proposed method contains three network modules – detection network, prior estimation network (PEN) and residual feature recovery block (RFRB).

Fig. 8.2 gives an overview of the proposed model. During source training, a source image (clean image) is passed to the detection network and the weights are learned by minimizing the detection loss, as shown in Fig. 8.2 with the source pipeline. For target training, a target image (weather-affected image) is forwarded through the network as shown in Fig. 8.2 by the target pipeline. As discussed earlier, weather-based degradations cause distortions in the feature space for the target images. In an attempt to de-distort these features, we introduce a set of residual feature recovery blocks in the target pipeline as shown in Fig. 8.2. This model is inspired from residual transfer framework proposed in [70] and is used to model residual features. The proposed PEN aids the detection network in adapting to the target domain by providing feedback through adversarial training using the proposed prior adversarial loss. In the following subsections, we briefly review the backbone network, followed by a discussion on the proposed prior-adversarial loss and residual feature recovery blocks.

8.2.1 Detection network

Following the existing domain adaptive detection approaches [16, 107, 121], we base our method on the Faster-RCNN [102] framework. Faster-RCNN is among the first end-to-end CNN-based object detection methods and uses anchor-based strategy to perform detection and classification.For simplicity, we decompose the Faster-RCNN network into three network modules: feature extractor network ($\mathcal{F}$), region proposal network (RPN) stage and region

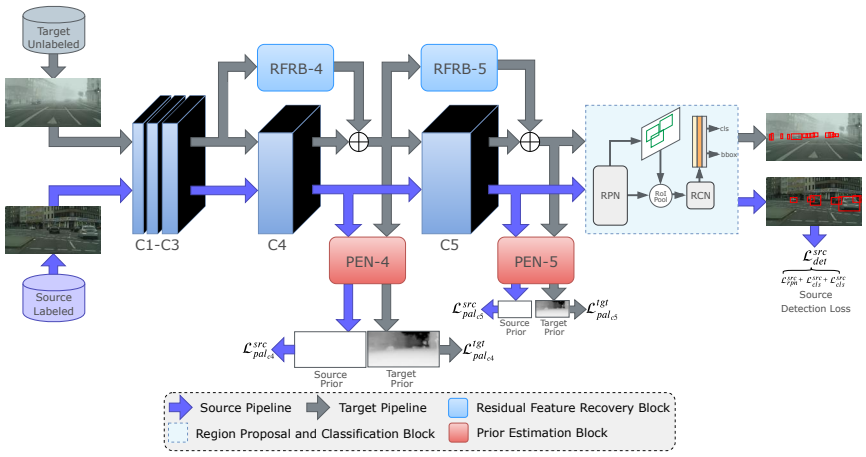


Figure 8.2: Overview of the proposed adaptation method. We use prior adversarial loss to supervise the domain discriminators. For the source pipeline, additional supervision is provided by detection loss. For target pipeline, feed-forward through the detection network is modified by the residual feature recovery blocks.

classification network (RCN). The arrangement of these modules are shown in the Fig. 8.2 with VGG model architecture as base network. Here, the feature extractor network consists of first five conv blocks of VGG and region classification network module is composed of fully connected layers of VGG. The region proposal network uses output of feature extractor network to generate a set of candidate object regions in a class agnostic way. Features corresponding to these candidates are pooled from the feature extractor and are forwarded through the region classification network to get the object classifications and bounding box refinements. Since we have access to the source domain images and their corresponding ground truth, these networks are trained to perform detection on the source domain by minimizing the following loss function,

$$\min_{\mathcal{F},\,\mathcal{G}} \quad \mathcal{L}_{det}^{src}\,, \quad \text{where} \tag{8.1}$$

$$\mathcal{L}_{det}^{src} \;=\; \mathcal{L}_{rpn}^{src} + \mathcal{L}_{bbox}^{src} + \mathcal{L}_{rcn}^{src}. \tag{8.2}$$

Here, $\mathcal{G}$ represents both region proposal and region classification networks, $\mathcal{L}_{rpn}^{src}$ denotes the region proposal loss, $\mathcal{L}_{bbox}^{src}$ denotes the bounding-box regression loss and $\mathcal{L}_{rcn}^{src}$ denotes the region classification loss. The details of these individual loss components can be found in [102].

8.2.2 Prior-adversarial training

As discussed earlier, weather-affected images, contain domain specific information. These images typically follow mathematical models of image degradation (see Eq. 8.8 and Eq. 8.9). We refer to this domain specific information as a *prior*. Detailed discussion about prior for haze and rain is provided later in the section. We aim to exploit these priors about the weather domain to better adapt the detector for weather affected images. To achieve that, we propose a prior-based adversarial training approach using prior estimation network (PEN) and prior adversarial loss (PAL).

Let $\mathcal{P}_l$ be PEN module introduced after the l^{th} conv block of $\mathcal{F}$ and let Z_{il}^{src} be the corresponding domain specific prior for any image, $x_i^s \in \mathcal{S}$. Then the PAL for the source domain is defined as follows,

$$\mathcal{L}_{pal_{cl}}^{src} = \frac{1}{n_s UV} \sum_{i=1}^{n_s} \sum_{j=1}^{U} \sum_{k=1}^{V} (Z_{il}^{src} - \mathcal{P}_l(\mathcal{F}_l(x_i^s)))_{jk}^2, \tag{8.3}$$

where, U and V are height and width of domain specific prior Z_{il}^{src} and output

feature $\mathcal{F}_l(x_i^s)$. Z_{il}^{src} denotes the source image prior, scaled down from image-level prior to match the scale at l^{th} conv block. Similarly, PAL for the target domain images, $x_i^t \in \mathcal{T}$, with the corresponding prior Z_{il}^{tgt} can be defined as,

$$\mathcal{L}_{pal_{cl}}^{tgt} = \frac{1}{n_t UV} \sum_{i=1}^{n_t} \sum_{j=1}^{U} \sum_{k=1}^{V} (Z_{il}^{tgt} - \mathcal{P}_l(\mathcal{F}_l(x_i^t)))_{jk}^2, \tag{8.4}$$

where, we apply PAL after conv4 (l=4) and conv5 (l=5) block (as shown in Fig. 8.2). Hence, the final source and target adversarial losses can be given as,

$$\mathcal{L}_{pal}^{src} = \frac{1}{2}(\mathcal{L}_{pal_{c5}}^{src} + \mathcal{L}_{pal_{c4}}^{src}), \tag{8.5}$$

$$\mathcal{L}_{pal}^{tgt} = \frac{1}{2}(\mathcal{L}_{pal_{c5}}^{tgt} + \mathcal{L}_{pal_{c4}}^{tgt}). \tag{8.6}$$

The prior estimation networks ($\mathcal{P}_5$ and $\mathcal{P}_4$) predict the weather-specific prior from the features extracted from $\mathcal{F}$. However, the feature extractor network $\mathcal{F}$ is trained to fool the PEN modules by producing features that are weather-invariant (free from weather-specific priors) and prevents the PEN modules from correctly estimating the weather-specific prior. Since, this type of training includes prior prediction and is also reminiscent of the adversarial learning used in domain adaptation, we term this loss as prior-adversarial loss. At convergence, the feature extractor network $\mathcal{F}$ should have devoid itself from any weather-specific information and as a result both prior estimation networks $\mathcal{P}_5$ and $\mathcal{P}_4$ should not be able to correctly estimate the prior. *Note that our goal at convergence is not to estimate the correct prior, but rather to learn weather-invariant features so that the detection network is able to generalize well to the target domain*. This training procedure can be expressed as the following

optimization,

$$\max_{\mathcal{F}} \min_{\mathcal{P}} \quad \mathcal{L}_{pal}^{src} + \mathcal{L}_{pal}^{tgt}. \tag{8.7}$$

Furthermore, in the conventional domain adaptation, a single label is assigned for entire target image to train the domain discriminator. By doing this, it is assumed that the entire image has undergone a constant domain shift. However this is not true in the case of weather-affected images, where degradations vary spatially. In such cases, the assumption of constant domain shift leads to incorrect alignment especially in the regions of minimal degradations. Incorporating the weather-specific priors overcomes this issue as these priors are spatially varying and are directly correlated with the amount of degradations. Hence, utilizing the weather-specific prior results in better alignment.

8.2.2.1 Haze prior

The effect of haze on images has been extensively studied in the literature [2, 27, 38, 63, 140, 142, 143]. Most existing image dehazing methods rely on the atmospheric scattering model for representing image degradations under hazy conditions and is defined as,

$$I(z) = J(z)t(z) + A(z)(1 - t(z)), \tag{8.8}$$

where I is the observed hazy image, J is the true scene radiance, A is the global atmospheric light, indicating the intensity of the ambient light, t is the transmission map and z is the pixel location. The transmission map is a

distance-dependent factor that affects the fraction of light that reaches the camera sensor. When the atmospheric light A is homogeneous, the transmission map can be expressed as $t(z) = e^{-\beta d(z)}$, where β represents the attenuation coefficient of the atmosphere and d is the scene depth.

Typically, existing dehazing methods first estimate the transmission map and the atmospheric light, which are then used in Eq. (8.8) to recover the observed radiance or clean image. The transmission map contains important information about the haze domain, specifically representing the light attenuation factor. We use this transmission as a domain prior for supervising the prior estimation (PEN) while adapting to hazy conditions. *Note that no additional human annotation efforts are required for obtaining the haze prior.*

8.2.2.2 Rain prior

Similar to dehazing, image deraining methods [61,62,137,139,141] also assume a mathematical model to represent the degradation process and is defined as follows,

$$I(z) = J(z) + R(z), \tag{8.9}$$

where I is the observed rainy image, J is the desired clean image, and R is the rain residue. This formulation models rainy image as a superposition of the clean background image with the rain residue. The rain residue contains domain specific information about the rain for a particular image and hence, can be used as a domain specific prior for supervising the prior estimation network (PEN) while adapting to rainy conditions. *Similar to the haze, we avoid the use of expensive human annotation efforts for obtaining the rain prior.*

In both cases discussed above (haze prior and rain prior), we do not use any ground-truth labels to estimate respective priors. Hence, our overall approach still falls into the category of unsupervised adaptation. Furthermore, these priors can be pre-computed for the training images to reduce the computational overhead during the learning process. Additionally, the prior computation is not required during inference and hence, the proposed adaptation method does not result in any computational overhead.

8.2.3 Residual Feature Recovery Block (RFRB)

As discussed earlier, weather-degradations introduce distortions in the feature space. In order to aid the de-distortion process, we introduce a set of residual feature recovery blocks (RFRBs) in the target feed-forward pipeline. This is inspired from the residual transfer network method proposed in [70]. Let $\Delta\mathcal{F}_l$ be the residual feature recovery block at the l^{th} conv block. The target domain image feedforward is modified to include the residual feature recovery block. For $\Delta\mathcal{F}_l$ the feed-forward equation at the l^{th} conv block can be written as,

$$\hat{\mathcal{F}}_l(x_i^t) \;=\; \mathcal{F}_l(x_i^t) \;+\; \Delta\mathcal{F}_l(\mathcal{F}_{l-1}(x_i^t)), \tag{8.10}$$

where, $\mathcal{F}_l(x_i^t)$ indicates the feature extracted from the l^{th} conv block for any image x_i^t sampled from the target domain using the feature extractor network $\mathcal{F}$, $\Delta\mathcal{F}_l(\mathcal{F}_{l-1}(x_i^t))$ indicates the residual features extracted from the output $l-1^{th}$ conv block, and $\hat{\mathcal{F}}_l(x_i^t)$ indicates the feature extracted from the l^{th} conv block for any image $x_i^t \in \mathcal{T}$ with RFRB modified feedforward. The RFRB modules are also illustrated in Fig. 8.2, as shown in the target feedforward

pipeline. It has no effect on source feedforward pipeline. In our case, we utilize RFRB at both conv4 ($\Delta\mathcal{F}_4$) and conv5 ($\Delta\mathcal{F}_5$) blocks. Additionally, the effect of residual feature is regularized by enforcing the norm constraints on the residual features. The regularization loss for RFRBs, $\Delta\mathcal{F}_4$ and $\Delta\mathcal{F}_5$ is defined as,

$$\mathcal{L}_{reg} = \frac{1}{n_t}\sum_{i=1}^{n_t}\sum_{l=4,5} \|\Delta\mathcal{F}_l(\mathcal{F}_{l-1}(x_i^t))\|_1, \tag{8.11}$$

8.2.4 Overall loss

The overall loss for training the network is defined as,

$$\max_{\mathcal{P}} \min_{\mathcal{F},\Delta\mathcal{F},\mathcal{G}} \quad \mathcal{L}_{det}^{src} - \mathcal{L}_{adv} + \lambda\mathcal{L}_{reg}, \quad \text{where} \tag{8.12}$$

$$\mathcal{L}_{adv} = \frac{1}{2}(\mathcal{L}_{pal}^{src} + \mathcal{L}_{pal}^{tgt}). \tag{8.13}$$

Here, $\mathcal{F}$ represents the feature extractor network, $\mathcal{P}$ denotes both prior estimation network employed after conv4 and conv5 blocks, i.e., $\mathcal{P}=\{\mathcal{P}_5, \mathcal{P}_4\}$, and $\Delta\mathcal{F}=\{\Delta\mathcal{F}_4, \Delta\mathcal{F}_5\}$ represents RFRB at both conv4 and conv5 blocks. Also, $\mathcal{L}_{det}^{src}$ is the source detection loss, $\mathcal{L}_{reg}$ is the regularization loss, and $\mathcal{L}_{adv}$ is the overall adversarial loss used for prior-based adversarial training.

8.3 Experiments and results

8.3.1 Implementation details

We follow the training protocol of [16, 107] for training the Faster-RCNN network. The backbone network for all experiments is VGG16 network [123]. We model the residuals using RFRB for the convolution blocks C4 and C5 of

the VGG16 network. The PA loss is applied to only these conv blocks modeled with RFRBs. The PA loss is designed based on the adaptation setting (Haze or Rain). The parameters of the first two conv blocks are frozen similar to [16,107]. The detailed network architecture for RFRBs, PEN and the discriminator are provided in supplementary material. During training, we set shorter side of the image to 600 with ROI alignment. We train all networks for 70K iterations. For the first 50K iterations, the learning rate is set equal to 0.001 and for the last 20K iterations it is set equal to 0.0001. We report the performance based on the trained model after 70K iterations. We set λ equal to 0.1 for all experiments.

In addition to comparison with recent methods, we also perform an ablation study where we evaluate the following configurations to analyze the effectiveness of different components in the network. Note that we progressively add additional components which enables us to gauge the performance improvements obtained by each of them,

- **FRCNN:** Source only baseline experiment where Faster-RCNN is trained on the source dataset.
- **FRCNN+D_5:** Domain adaptation baseline experiment consisting of Faster-RCNN with domain discriminator after conv5 supervised by the domain adversarial loss.
- **FRCNN+D_5+R_5:** Starting with FRCNN+D_5 as the base configuration, we add an RFRB block after conv4 in the Faster-RCNN. This experiment enables us to understand the contribution of the RFRB block.
- **FRCNN+P_5+R_5:** We start with FRCNN+D_5+R_5 configuration and replace domain discriminator and domain adversarial loss with prior estimation

network (PEN) and prior adversarial loss (PAL). With this experiment, we show the importance of training with the proposed prior-adversarial loss.

- **FRCNN+P_{45}+R_{45}:** Finally, we perform the prior-based feature alignment at two scales: conv4 and conv5. Starting with FRCNN+P_5+R_5 configuration, we add an RFRB block after conv3 and a PEN module after conv4. This experiment corresponds to the configuration depicted in Fig. 8.2. This experiment demonstrates the efficacy of the overall method in addition to establishing the importance of aligning features at multiple levels in the network.

Following the protocol set by the existing methods [16, 107, 121], we use mean average precision (mAP) scores for performance comparison.

8.3.2 Adaptation to hazy conditions

In this section, we present the results corresponding to adaptation to hazy conditions on the following datasets: (i) Cityscapes $\rightarrow$ Foggy-Cityscapes [109], (ii) Cityscapes $\rightarrow$ RTTS [60], and (iii) WIDER [135] $\rightarrow$ UFDD-Haze [79]. In the first two experiments, we consider Cityscapes [17] as the source domain. Note that the Cityscapes dataset contains images captured in clear weather conditions.

Cityscapes $\rightarrow$ Foggy-Cityscapes: In this experiment, we adapt from Cityscapes to Foggy-Cityscapes [109]. The Foggy-Cityscapes dataset was recently proposed in [109] to study the detection algorithms in the case of hazy weather

conditions. Foggy-Cityscapes is derived from Cityscapes dataset by simulating fog on the clear weather images of Cityscapes. Both Cityscapes and Foggy-Cityscapes have the same number of categories which include, car, truck, motorcycle/bike, train, bus, rider and person. Similar to [16], [107], we utilize 2975 images of both Cityscapes and Foggy-Cityscapes for training. Note that we use annotations only from the source dataset (Cityscapes) for training the detection pipeline. For evaluation we consider a non overlapping validation set of 500 images provided by the Foggy-Cityscapes dataset.

We compare the proposed method with two categories of approaches: *(i) Dehaze+Detect:* Here, we employ dehazing network as pre-processing step and perform detection using Faster-RCNN trained on source (clean) images. For pre-processing, we chose two recent dehazing algorithms: DCPDN [140] and Grid-Dehaze [68]. *(i) DA-based methods:* Here, we compare with following recent domain-adaptive detection approaches: DA-Faster [16], SWDA [107], DiversifyMatch [49], Mean Teacher with Object Relations (MTOR) [10], Selective Cross-Domain Alignment (SCDA) [148] and Noisy Labeling [48]. The corresponding results are presented in Table 8.1.

It can be observed from Table 8.1, that the performance of source-only training of Faster-RCNN is in general poor in the hazy conditions. Adding DCPDN and Gird-Dehaze as preprocessing step improves the performance by $\sim$2% and $\sim$4%, respectively. Compared to the domain-adaptive detection approaches, pre-processing + detection results in lower performance gains. This is because even after applying dehazing there still remains some domain shift. Hence, using adaptation would be a better approach for mitigating the domain

(a) (b)

Figure 8.3: Detection results on Foggy-Cityscapes. (a) DA-Faster RCNN [16]. (b) Proposed method. The bounding boxes are colored based on the detector confidence. DA-Faster-RCNN produces detections with low confidence in addition to missing the truck class. Our method is able to output high confidence detections without missing any objects.

Table 8.1: Performance comparison for the Cityscapes → Foggy-Cityscapes experiment.

Method		prsn	rider	car	truc	bus	train	bike	bcycle	mAP
Baseline	FRCNN [102]	25.8	33.7	35.2	13.0	28.2	9.1	18.7	31.4	24.4
Dehaze	DCPDN [140]	27.9	36.2	35.2	16.0	28.3	10.2	24.6	32.5	26.4
	Grid-Dehaze [68]	29.7	40.4	40.3	21.3	30.0	9.1	25.6	36.7	29.2
DA-Methods	DAFaster [16]	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
	SCDA [148]	33.5	38.0	48.5	26.5	39.0	23.3	28.0	33.6	33.8
	SWDA [107]	29.9	42.3	43.5	24.5	36.2	32.6	30.0	35.3	34.3
	DM [49]	30.8	40.5	44.3	**27.2**	38.4	34.5	28.4	32.2	34.6
	MTOR [10]	30.6	41.4	44.0	21.9	38.6	**40.6**	28.3	35.6	35.1
	NL [48]	35.1	42.1	49.2	30.1	45.3	26.9	26.8	36.0	36.5
Ours	FRCNN+D_5	30.9	38.5	44.0	19.6	32.9	17.9	24.1	32.4	30.0
	FRCNN+D_5+R_5	32.8	44.7	49.9	22.3	31.7	17.3	26.9	37.5	32.9
	FRCNN+P_5+R_5	33.4	42.8	50.0	24.2	40.8	30.4	33.1	37.5	36.5
	FRCNN+P_{45}+R_{45}	**36.4**	**47.3**	**51.7**	22.8	**47.6**	34.1	**36.0**	**38.7**	**39.3**

shift. Here, the use of simple domain adaptation [30] (FRCNN+D_5) improves the source-only performance. The addition of $RFRB_5$ (FRCNN+D_5+R_5) results in further improvements, thus indicating the importance of RFRB blocks. However, the conventional domain adaptation loss assumes constant domain

shift across the entire image, resulting in incorrect alignment. The use of prior-adversarial loss (FRCNN+P_5+R_5) overcomes this issue. We achieved 3.6% improvement in overall mAP scores, thus demonstrating the effectiveness of the proposed prior-adversarial training. Note that, FRCNN+P_5+R_5 baseline achieves comparable performance with state-of-the-art. Finally, by performing prior-adversarial adaptation at an additional scale (FRCNN+P_{45}+R_{45}), we achieve further improvements which surpasses the existing best approach [48] by 2.8%. Fig. 8.3 shows sample qualitative detection results corresponding to the images from Foggy-Cityscapes. Results for the proposed method are compared with DA-Faster-RCNN [16]. It can be observed that the proposed method is able to generate comparatively high quality detections.

We summarize our observations as follows: (i) Using dehazing as a pre-processing step results in minimal improvements over the baseline Faster-RCNN. Domain adaptive approaches perform better in general. (ii) The proposed method outperforms other methods in the overall scores while achieving the best performance in most of the classes. See supplementary material for more ablations.

Cityscapes $\rightarrow$ RTTS: In this experiment, we adapt from Cityscapes to the RTTS dataset [60] . RTTS is a subset of a larger RESIDE dataset [60], and it contains 4,807 unannotated and 4,322 annotated real-world hazy images covering mostly traffic and driving scenarios. We use the unannotated 4,807 images for training the domain adaptation process. The evaluation is performed on

Table 8.2: Performance comparison for the Cityscapes → RTTS experiment.

Method		prsn	car	bus	bike	bcycle	mAP
Baseline	FRCNN [102]	46.6	39.8	11.7	19.0	37.0	30.9
Dehaze	DCPDN [140]	**48.7**	39.5	12.9	19.7	37.5	31.6
	Grid-Dehaze [68]	29.7	25.4	10.9	13.0	21.4	20.0
DA	DAFaster [16]	37.7	48.0	14.0	**27.9**	36.0	32.8
	SWDA [107]	42.0	46.9	15.8	25.3	37.8	33.5
Ours	Proposed	37.4	**54.7**	**17.2**	22.5	**38.5**	**34.1**

Table 8.3: Results (mAP) of the adaptation experiments from WIDER-Face to UFDD Haze and Rain.

Method	UFDD-Haze	UFDD-Rain
FRCNN [102]	46.4	54.8
DAFaster [16]	52.1	58.2
SWDA [107]	55.5	60.0
Proposed	**58.5**	**62.1**

the annotated 4,322 images. RTTS has total five categories, namely motorcycle/bike, person, bicycle, bus and car. This dataset is the largest available dataset for object detection under real world hazy conditions.

In Table 8.2, the results of the proposed method are compared with Faster-RCNN [102], DA-Faster [16] and SWDA [107] and the dehaze+detection baseline as well. For RTTS dataset, the pre-processing with DCPDN improves the Faster-RCNN performance by ∼1%. Surprisingly, Grid-Dehaze does not help the Faster-RCNN baseline and results in even worse performance. Whereas, the proposed method achieves an improvement of 3.1% over the baseline Faster-RCNN (source-only training), while outperforming the other recent methods.

WIDER-Face → UFDD-Haze: Recently, Nada *et al.* [79] published a benchmark face detection dataset which consists of real-world images captured

Table 8.4: Performance comparison for the Cityscapes → Rainy-Cityscapes experiment.

Method		prsn	rider	car	truc	bus	train	bike	bcycle	mAP
Baseline	FRCNN	21.6	19.5	38.0	12.6	30.1	24.1	12.9	15.4	21.8
Derain	DDN [29]	27.1	30.3	50.7	23.1	39.4	18.5	21.2	24.0	29.3
	SPANet [132]	24.9	28.9	48.1	21.4	34.8	16.8	17.6	20.8	26.7
DA	DAFaster [16]	26.9	28.1	50.6	23.2	39.3	4.7	17.1	20.2	26.3
	SWDA [107]	29.6	**38.0**	52.1	27.9	**49.8**	28.7	24.1	25.4	34.5
Ours	FRCNN+D_5	29.1	34.8	52.0	22.0	41.8	20.4	18.1	23.3	30.2
	FRCNN+D_5+R_5	28.8	33.1	51.7	22.3	41.8	24.9	22.2	24.6	31.2
	FRCNN+P_5+R_5	29.7	34.3	52.5	23.6	47.9	32.5	24.0	25.5	33.8
	FRCNN+P_{45}+R_{45}	**31.3**	34.8	**57.8**	**29.3**	48.6	**34.4**	**25.4**	**27.3**	**36.1**

under different weather-based conditions such as haze and rain. Specifically, this dataset consists of 442 images under the haze category. Since, face detection is closely related to the task of object detection, we evaluate our framework by adapting from WIDER-Face [135] dataset to UFDD-Haze dataset. WIDER-Face is a large-scale face detection dataset with approximately 32,000 images and 199K face annotations. The results corresponding to this adaptation experiment are shown in Table 8.3. It can be observed from this table that the proposed method achieves better performance as compared to the other methods.

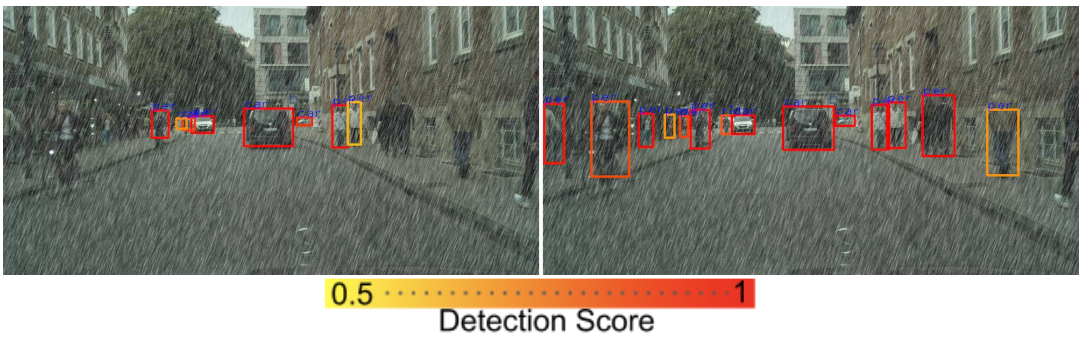
8.3.3 Adaptation to rainy conditions

In this section, we present the results of adaptation to rainy conditions. Due to lack of appropriate datasets for this particular setting, we create a new rainy dataset called Rainy-Cityscapes and it is derived from Cityscapes. It has the same number of images for training and validation as Foggy-Cityscapes. First, we discuss the simulation process used to create the dataset, followed by a discussion of the evaluation and comparison of the proposed method with

other methods.

Rainy-Cityscapes: Similar to Foggy-Cityscapes, we use a subset of 3475 images from Cityscapes to create synthetic rain dataset. Using [1], several masks containing artificial rain streaks are synthesized. The rain streaks are created using different Gaussian noise levels and multiple rotation angles between 70° and 110°. Next, for every image in the subset of the Cityscapes dataset, we pick a random rain mask and blend it onto the image to generate the synthetic rainy image. More details and example images are provided in supplementary material.

Cityscapes → Rainy-Cityscapes: In this experiment, we adapt from Cityscapes to Rainy-Cityscapes. We compare the proposed method with recent methods such as DA-Faster [16] and SWDA [107]. Additionally, we also evaluate performance of two derain+detect baselines, where state of the art methods such as DDN [29] and SPANet [132] are used as a pre-processing step to the Faster-RCNN trained on source (clean) images. From the Table 8.4 we observe that such methods provide reasonable improvements over the Faster-RCNN baseline. However, the performance gains are much lesser as compared to adaptation methods, for the reasons discussed in the earlier sections (Sec. 8.3.2). Also, it can be observed from Table 8.4, that the proposed method outperforms the other methods by a significant margin. Additionally, we present the results of the ablation study consisting of the experiments listed in Sec. 8.3.1. The introduction of domain adaptation loss significantly improves



(a) (b)

Figure 8.4: (a) DA-Faster RCNN [16]. (b) Proposed method. The bounding boxes are colored based on the detector confidence. DA-Faster-RCNN misses several objects. Our method is able to output high confidence detections without missing any objects.

the source only Faster-RCNN baseline, resulting in approximately 9% improvement for FRCNN+D_5 baseline in Table 8.4. This performance is further improved by 1% with the help of residual feature recovery blocks as shown in FRCNN+D_5+R_5 baseline. When domain adversarial training is replaced with prior adversarial training with PAL, i.e. FRCNN+P_5+R_5 baseline, we observe 2.5% improvements, showing effectiveness of the proposed training methodology. Finally, by performing prior adversarial training at multiple scales, the proposed method FRCNN+P_{45}+R_{45} observes approximately 2% improvements and also outperforms the next best method SWDA [107] by 1.6%. Fig. 8.4 illustrates sample detection results obtained using the proposed method as compared to a recent method [16]. The proposed method achieves superior quality detections.

WIDER-Face → UFDD-Rain: In this experiment, we adapt from WIDER-Face to UFDD-Rain [79]. The UFDD-Rain dataset consists of 628 images collected under rainy conditions. The results of the proposed method as compared to

the other methods are shown in Table 8.3. It can be observed that the proposed method outperforms the source only training by 7.3%.

8.3.4 Conclusion

We addressed the problem of adapting object detectors to hazy and rainy conditions. Based on the observation that these weather conditions cause degradations that can be mathematically modeled and cause spatially varying distortions in the feature space, we propose a novel prior-adversarial loss that aims at producing weather-invariant features. Additionally, a set of residual feature recovery blocks are introduced to learn residual features that can aid efficiently aid the adaptation process. The proposed framework is evaluated on several benchmark datasets such as Foggy-Cityscapes, RTTS and UFDD. Through extensive experiments, we showed that our method achieves significant gains over the recent methods in all the datasets.

Chapter 9

Federated Learning-based User Authentication

9.1 Federated average vs non-IID data

We illustrate the challenges in detail by considering a case study with FedAvg algorithm to show how the performance changes when the IID assumption of the FL/SL framework does not hold. Subsequently, we discuss the proposed solution FAA which overcomes these challenges to provide an improved user authentication system.

The key challenge in Federated Active Authentication (FAA) is the non-IID nature of the data distribution across mobile devices. This issue directly affects the user authentication performance. First, let us briefly discuss the definition of independent and identically distributed data. In the context of federated learning, when data is said to be distributed in an IID manner, it means that each device has equal number of data samples from all users. This is the most common assumption in federated learning and is very crucial to train a model using the FedAvg algorithm. To show how deviation from this assumption

affects the performance of FedAvg algorithm, let us quantify the *IID-ness* of the data distributed among devices. Let us assume that there are N devices containing data from K users. Let K_i be the number of users contained in the i^{th} device dataset having sufficient number of data samples. Let $qIID$ denote the quantification of "IID-ness" of the distributed data in the federated framework. For simplicity, let us assume that each device has equal number of data samples. Given these assumptions, the $qIID$ can be formally written as:

$$qIID = \frac{\frac{1}{N}\sum_{i=1}^{N}\frac{K_i}{K} - \frac{1}{K}}{1 - \frac{1}{K}}, \tag{9.1}$$

where $qIID = 1$ when the data distribution across devices is the most IID and it decreases as the distribution deviates from IID. The value $qIID = 0$ represents the most non-IID data distribution across devices. The proposed FAA problem operates on a specific value of $qIID = 0$, where the number of devices are equal to the number of users, i.e., $N = K$. To show how the

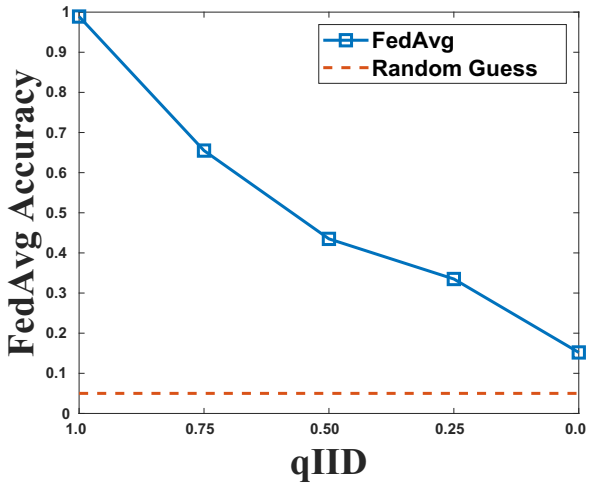


Figure 9.1: Performance of federated averaging (FedAvg) algorithm with varying value of $qIID$ representing the way data is distributed among devices.

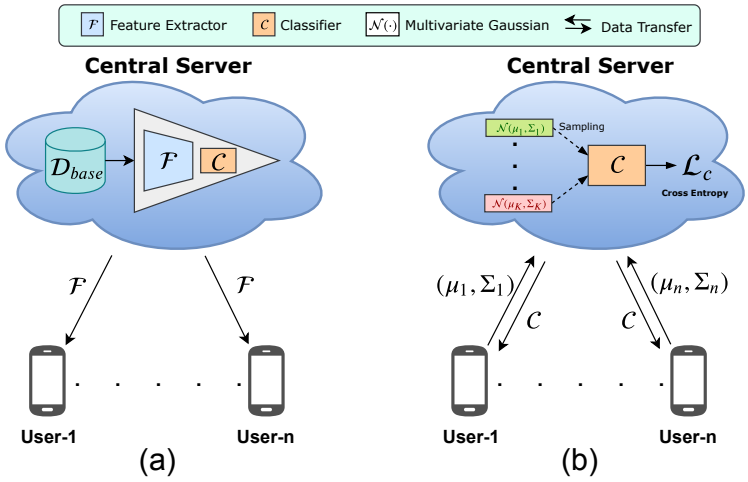


Figure 9.2: Block diagram describing the training of the proposed method for federated active authentication. (a) Step-1 of the proposed method: training a model on base dataset, (b) Step-2 & Step-3 of proposed method: local mobile devices compute feature statistics of the user, the central server trains a classifier using these statistics, the classifier is then sent to the individual devices and together with the feature extractor, it is used as an authentication model.

performance of FedAvg algorithm changes when the IID assumption is violated, we perform identification experiments using the UMDAA-01 dataset by changing the $qIID$ value from one to zero. As evident from Fig. 9.1, the FedAvg performance heavily relies on the IID assumption. The more distribution of data among devices in the federated learning framework deviates from the IID assumption, the performance of FedAvg degrades significantly. The reason for this reduction in performance is due to averaging of weights at the central server. This makes sense as the individual models are trained on the data with similar data distributions. Interestingly, for the case of federated active authentication, where the distribution of data among devices is the most non-IID, i.e., $qIID = 0$, the performance is almost close to random guessing baseline.

9.2 Federated active authentication

9.2.1 Proposed training methodology

Step-1. Let us first consider a randomly initialized deep network model $\mathcal{M}$ at the central server. Furthermore, let us denote a publicly available face recognition dataset as, $\mathcal{D}_{base} = \{x_i^{base}, y_i^{base}\}_{i=1}^{N_{base}}$. Here, x_i^{base} are the face images having corresponding labels y_i^{base} where the dataset contains a total of N_{base} images. Note that, $\mathcal{D}_{base}$ does not have any category that overlap with data available in the individual mobile devices. As shown in Fig. 9.2(a), the deep network model $\mathcal{M}$ is then trained at server side on the dataset $\mathcal{D}_{base}$ with the help of the following loss:

$$\mathcal{L}_{base} = \frac{1}{N_{base}} \sum_{i=1}^{N_{base}} \mathcal{L}_c(\mathcal{M}(x_i^{base}),\ y_i^{base}), \tag{9.2}$$

where, $\mathcal{L}_c$ is the cross-entropy loss function. Once the model $\mathcal{M}$ is trained, it is further divided into two networks, namely, feature extractor network ($\mathcal{F}$) and classifier network ($\mathcal{C}$). The central server sends feature extractor network $\mathcal{F}$ to all the mobile devices connected to the central server.

Step-2. Assume that there are K mobile devices (i.e. K users) and the ith device has the corresponding dataset $\mathcal{D}_i$ containing n_i face images of the user. All K devices are connected to the central server. With the help of network $\mathcal{F}$, each device estimates the feature mean and variance of the corresponding user, which we refer to as *user impressions*. For the ith device, the user impressions

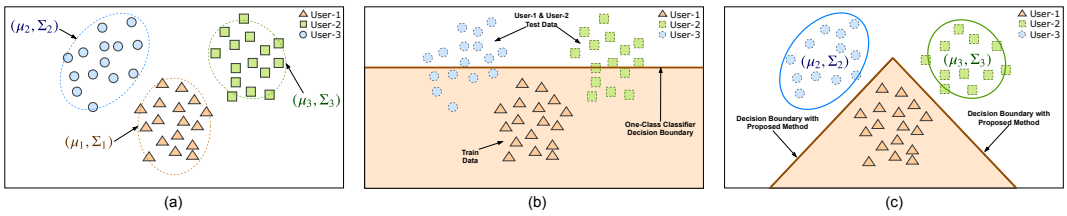


Figure 9.3: Toy example with three users to show the effectiveness of the proposed method compared to one-class modeling based methods. (a) Feature space location (mean μ_i) and shape (variance Σ_i) estimated for each user. (b) Modeling as a one-class classification problem to learn a decision boundary for user-1. When such a model is tested there are many samples from user-2 and user-3 that are mis-classified as user-1. (c) Learning a decision boundary using the proposed method to train the authentication model for user-1 using user-1, user-2 and user-3's mean and variance. This model does not make the same mistake of mis-classifying user-2 and user-3 data as user-1 similar to one-class based method. As can be seen from the figure, the learned decision boundary is also better in comparison to one-class method.

can be estimated as:

$$\begin{aligned} \mu_i &= \frac{1}{n_i} \sum_{x_j \in \mathcal{D}_i} \mathcal{F}(x_j), \\ \Sigma_i &= \frac{1}{n_i} \sum_{x_j \in \mathcal{D}_i} (\mathcal{F}(x_j) - \mu_i)(\mathcal{F}(x_j) - \mu_i)^T, \end{aligned} \tag{9.3}$$

where x_j is the jth face image in $\mathcal{D}_i$. Each user impression (μ_i, Σ_i), provides a reasonable estimate regarding the location and the shape of the ith user distribution in the feature space of network $\mathcal{F}$. Once all devices have finished estimating user impressions, they are sent to the central server, which creates a Gaussian approximated feature space model of each user as $\mathcal{N}(\mu_i, \Sigma_i)$. This approximation is inspired by the work of Seddik *et al.* [118], which showed that the feature space of deep networks can be well approximated with only first and second order statistics of the features.

Step-3. With the help of Gaussian approximated feature space models of all users, we create a combined dataset as, $\mathcal{D} = \{f_j \sim \mathcal{N}(\mu_i, \Sigma_i), y_j = i\}$. We

make sure that each user has exactly M number of samples, resulting in total $K \times M$ samples. As shown in the Fig. 9.2(b), we fine-tune the identification network using the loss given as:

$$\mathcal{L} \;=\; \frac{1}{K \times M} \sum_{j=1}^{K \times M} \mathcal{L}_c(\mathcal{C}(f_j),\, y_j), \tag{9.4}$$

where, y_j is the corresponding user id of feature f_j and $\mathcal{L}_c$ is the cross-entropy loss. Once the classifier $\mathcal{C}$ is trained, its architecture and weights are sent to all mobile devices. Both $\mathcal{F}$ and $\mathcal{C}$ together form the authentication system. Furthermore, in Fig. 9.3 we illustrate how the proposed approach is able to utilize user-impressions to improve the authentication with the help of a toy example with three users. The current algorithms model the active authentication problem as one-class classification. Due to this, the classifier learned for a particular user still has some risk of failing to restrict the device access to other users, illustrated in Fig. 9.3(b). However, as shown in Fig. 9.3(c), the proposed approach is able to utilize user impressions from other users to learn a more compact decision boundary and improve the authentication performance.

9.2.2 Testing

For any test face image x_j, we compute the authentication score corresponding to the ith user as,

$$S_j^i \;=\; \mathbb{I}_{[\tilde{y}_j = i]}\, \mathcal{H}[\mathcal{C}(\mathcal{F}(x_j))] \;+\; \mathbb{I}_{[\tilde{y}_j \neq i]}\, \mathcal{H}[q], \tag{9.5}$$

where, $\tilde{y}_j$ is the predicted label of the test image x_j, i.e., argmax $\mathcal{C}(\mathcal{F}(x_j))$. The $\mathbb{I}_{[c]}$ is an indicator function which is 1 when condition c is satisfied and

0 otherwise. Vector $\mathcal{C}(\mathcal{F}(x_j))$ is a $K \times 1$ prediction vector. The function $\mathcal{H}[\cdot]$ calculates the entropy of the input probability vector. The vector q is $K \times 1$ probability vector with $q_1 = q_2 = ... = q_K = \frac{1}{K}$. When the predicted-id from the authentication model matches the user-id, the first term assigns the score S_j^i as the entropy of the prediction vector, i.e., $\mathcal{C}(\mathcal{F}(x_j))$. When the predicted-id does not match the user-id, the second term penalizes the input for this misclassification by assigning high entropy value to the score S_j^i. When both the terms are added together they encode the score of an input image belonging to the authorized user. Higher score indicates potentially unauthorized user and vice versa.

9.3 Experiments and results

9.3.1 Implementation details

For all experiments, we utilize the VGG16 [123] trained on the VGGFace dataset [88]. We consider all conv blocks of VGG16 as the feature extractor $\mathcal{F}$ and all fully-connected layers as the classifier $\mathcal{C}$. The mean and variance for each user are estimated by flattening the output of $\mathcal{F}$, later used in the server to fine-tune $\mathcal{C}$. For training, we utilize SGD optimizer with learning rate 0.001 and momentum 0.9. We train till 100 epochs with the batch size of 64. For all methods, the hyper-parameters are selected based on a validation set. The performance of all methods is evaluated using the average detection accuracy (ADA), defined as:

$$ADA = 0.5 * (TPR + TNR),$$

(a) MOBIO (b) UMDAA-01 (c) UMDAA-02

Figure 9.4: Sample face images from the (a) MOBIO, (b) UMDAA-01 and (c) UMDAA-02 datasets.

where, TNR and TPR represent true negative rate and true positive rate, respectively.

9.3.2 Datasets

MOBIO. The MOBIO [35] dataset contains face and voice data from 150 individuals collected in six different sessions and locations. It is collected using smart phones and/or laptop. For experiments, we only consider the face data. Out of the three datasets, MOBIO is relatively easy as it contains only front facing face images captured in well-lit conditions. Sample images are shown in Fig. 9.4(a). The figures provide a reasonable illustration of the variations present in the dataset. For the experiment, we consider the first 75 individuals as the enrolled users and the remaining 75 individuals as unknown/unauthorized users. We create a 50/50 split of data for training and testing for all 150 individuals.

UMDAA-01. The UMDAA-01 [26] contains face images of 50 different individuals collected using iPhone 5s in three different sessions with varying lighting conditions. Apart from varying illumination conditions, the dataset also contains multiple other variability in the form of pose, occlusion, facial

Table 9.1: Performance comparison with state-of-the-art active authentication methods evaluated in terms of average detection accuracy. The best performing method for each dataset is shown in bold fonts.

	1SVM	k1SVM	SVDD	kSVDD	kNFST	1vSet	1MPM	DMPM	OC-ACNN	Proposed
MOBIO	0.632 (0.004)	0.748 (0.004)	0.582 (0.007)	0.763 (0.013)	0.560 (0.003)	0.670 (0.005)	0.768 (0.003)	0.825 (0.007)	0.938 (0.005)	**0.998 (0.003)**
UMDAA-01	0.622 (0.002)	0.731 (0.009)	0.615 (0.018)	0.701 (0.009)	0.567 (0.012)	0.593 (0.017)	0.816 (0.003)	0.869 (0.001)	0.891 (0.002)	**0.954 (0.005)**
UMDAA-02	0.614 (0.008)	0.649 (0.004)	0.515 (0.007)	0.550 (0.007)	0.556 (0.003)	0.538 (0.003)	0.722 (0.006)	0.760 (0.007)	0.735 (0.009)	**0.813 (0.006)**

expressions etc. Sample images are shown in Fig. 9.4(b). We consider the first 25 individuals as the enrolled users and the remaining 25 users as unknown/unauthorized. Similar to MOBIO, we create a 50/50 train-test split and use the train split for training.

UMDAA-02. The UMDAA-02 [73] contains information from 18 different sensors such as touch pattern, face images, accelerometer readings from 44 individuals collected using Nexus5 across two months. For this experiment, we only utilize face images of all users. As can be seen from Fig. 9.4(c), out of all three datasets, UMDAA-02 contains the most variability in the data samples, proving it to be the most challenging dataset. We consider the first 22 individuals as the enrolled users and the remaining as unknown.

9.3.3 Experiments

We consider the following methods from the active authentication literature for comparison:

1. Linear OCSVM (1SVM): One-class SVM (OC-SVM) as formulated in [113] is trained with a linear kernel on features of given user..

2. Linear SVDD (SVDD): Support vector data descriptor (SVDD) with a

linear kernel as formulated in [124] is trained on features of given user.

3. Kernel OCSVM (k1SVM): OC-SVM as formulated in [113] is trained on the given features with a radial basis function (RBF) kernel.

4. Kernel SVDD (kSVDD): SVDD with RBF kernel as formulated in [124] is trained on given features.

5. One-class kNFST (kNFST): Kernel null foley-sammon transform is used as proposed in [7]. kNFST finds a single null-space direction in feature space where intra-class distance of the class is low.

6. One-vs-set Machines (1vSet): As proposed in [5], two hyper-planes are optimized to enclose given category features within a slab in feature space.

7. Single-MPM (1MPM): Proposed in [32], 1MPM considers second order statistics to learn a better hyperplane that separates origin from the one-class data in the feature space.

8. Dual-MPM (DMPM): Proposed in [95], DMPM extends the 1MPM formulation by learning an additional hyperplane that better encloses given features.

9. OC-ACNN: Method proposed in [84], develop a deep convolutional neural network based one-class classifier by using Gaussian as pseudo-negative samples and regularizing the feature space with a decoder network.

Table. 9.1 compares the performance of the proposed method with the state-of-the-art active authentication models. Out of all methods, 1SVM's and SVDD's performances are the lowest. Both of these methods are able to improve the performance when the kernel trick is incorporated into their formulations as shown by k1SVM and kSVDD, respectively. 1vSet and kNFST prove competitive against the classical one-class formulations such as 1SVM

Table 9.2: Impact on average detection accuracy with increasing the number of unknown/unauthorized users for the UMDAA-01.

Number of Unknown User	10	15	20	25
UMDAA-01	0.983 (0.003)	0.976 (0.003)	0.963 (0.002)	0.954 (0.005)

and SVDD. Out of all the methods based on hyperplane optimization formulation, the MPM-based methods clearly outperform all the others. Specifically, DMPM is able to outperform 1MPM by $\sim$5%, $\sim$6% and $\sim$4%, respectively on MOBIO, UMDAA-01 and UMDAA-02 datasets. OC-ACNN provides a considerable improvement compared to DMPM on MOBIO and UMDAA-01, but under performs on UMDAA-02. The proposed method outperforms all the other methods. More precisely, the proposed method observes $\sim$6%, $\sim$6% and $\sim$5% improvement over the next best baseline on MOBIO, UMDAA-01 and UMDAA-02, respectively. This improvement can be largely attributed to the fact that federated learning framework enables privacy preserving collaboration among devices that results in a better active authentication system compared to the traditional one-class modeling based methods.

Impact of Number of Unknown. Table. 9.2 shows the impact of varying the number of unknown/unauthorized users on the authentication system. For the experiment, we consider the UMDAA-01 dataset with all the implementation detail kept the same as described in Sec. 9.3.1 and the number of enrolled users are fixed to 25. As evident from the table, the performance decreases as we increase the number of unknown/unauthorized user during testing.

9.3.4 Fedarated/split learning vs proposed method

We compare the performance of FL and SL approaches with the proposed method. We evaluate these methods on all three datasets using the experimental protocol described in Sec. 9.3.2. As can be seen from Fig. 9.5, the proposed method is able to perform much better compared to both FedAvg and Split Learning Approach (SLA) [36] on all three datasets. In the case of MOBIO, both FedAvg and SLA perform the best compared to the other two datasets, providing average detection accuracy of 61.2% and 92%, respectively. In comparison, the proposed approach is able to achieve 99.8% average detection accuracy, resulting in nearly 38% and 7% improvement on the MOBIO dataset, respectively. For the slightly challenging UMDAA-01 dataset, when the authentication model is trained using FedAvg and SLA, the model achieves the performance of 52.4% and 89%, respectively. Compared to FedAvg and SLA, the proposed approach achieves 95.4% average detection accuracy. Similarly,

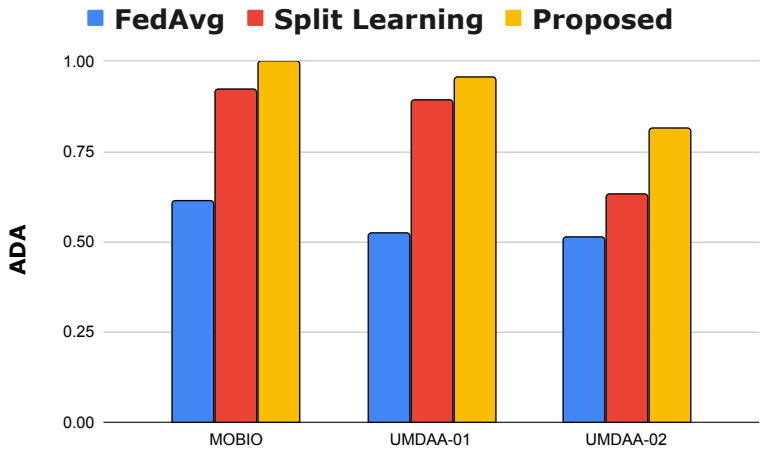


Figure 9.5: Comparing the performance between FedAvg, Split Learning [36] and the proposed method on the MOBIO, UMDAA-01 and UMDAA-02 datasets.

FedAvg and SLA perform about 51% and 62%, respectively on the most challenging UMDAA-02 dataset. Whereas, the proposed approach achieves 81.2% average detection accuracy, resulting in respective 29% and 19% improvement. As discussed in Sec. 9.1, the major reason why FL/SL methods perform poorly is due to the highly non-IID nature (i.e. $qIID = 0$) of the federated active authentication problem. Though SLA comes very close to the performance of the proposed method, it still requires multiple rounds of communication between device and server. In contrast, the proposed approach requires only one round of communication between device and server.

9.3.5 Conclusion

We proposed a novel approach for user active authentication based on federated and split learning frameworks, called Federate Active Authentication. We point out the limitations of existing active authentication methods that model it as a one-class classification problem. The proposed method utilizes the federated/split learning framework to go beyond the one-class assumption for user active authentication. We also show that existing federated/split learning algorithms perform poorly on the federated active authentication setting. To address these issues, we proposed a novel method that extracts feature statistics of each user and trains a classification network to perform a multi-class classification, resulting in an efficient training strategy and improved authentication model. The proposed method is evaluated on three publicly available datasets and it is shown that it can perform better compared to both one-class modeling based active authentication methods and

existing federated/split learning approaches. Furthermore, we analyze the effectiveness of the proposed method under varying number of enrolled and unknown/unauthorized users.

Chapter 10

Conclusion and Future Work

In this thesis, we attempted to address two key issues with existing visual recognition systems, namely, ability to detect unknown/novel instances during testing, and ability to generalize to novel visual domains without supervision. These two factors are very critical to a visual systems' real-world performance and we explore different techniques to overcome respective challenges. Specifically, we explore the problem of detecting unknown instances in both one-class and multi-class setting. For one-class setting, we show that it is possible to perform an end-to-end training of convolutional neural networks to learn better representations. In the end, we also showed how such a system can be improved with the help of federated learning. For multi-class setting, we explored the use of patch-level activity patterns and their role in understanding network behavior under known/unknown category inputs. We showed that such patch-level activity information is very useful in identifying whether the input test image belongs to a known or unknown category. We proposed a novel training strategy that utilizes such patch-level activity information and improves the unknown detection ability of visual recognition

systems.

Additionally, we study the generalization capability of visual recognition systems under the dataset distribution shift/domain shift. We explore this domain shift problem for the task of both multi-class novelty detection and general object detection. We showed the limited ability of existing novelty detection methods in generalizing to novel visual domains and proposed an approach that tackles this issue with the help of generative adversarial networks. For the task of object detection, we specifically consider the issue of adverse weather conditions where the detection performance suffers when images are degraded by haze/rain. We show that leveraging additional domain information extracted from the mathematical models of hazy/rainy conditions can guide the adaptation of object detectors and help improve the performance.

10.1 Future research directions

In this thesis, we explored multiple solutions that tackled many challenges related to unknown instance detection and domain adaptation. However, there is still more work required in other challenges related to these problems:

- There is a need to explore unknown instance detection for more challenging datasets that better mimic the real-world conditions.
- The security of one-class novelty detection models against adversarial attacks has not been explored, which will be an important real world challenge for the deployment of these systems.
- It is also important to make unknown instance detection systems more

explainable. This would help in improving our understanding of what aspects are important in identifying known/unknown categories.

- The domain adaptive object detection methods have only focused on unsupervised adaptation scenario. Exploring real world constraints like open-set/partial/universal domain adaptation, semi-supervised/weakly-supervised/test-time adaptation, class-imbalance issues, source-free/multi-source/multi-target conditions would further improve the real-world deployability of these models.

Bibliography

[1] https://www.photoshopessentials.com/photo-effects/photoshop-weather-effects-rain/.

[2] Cosmin Ancuti, Codruta O Ancuti, and Radu Timofte. Ntire 2018 challenge on image dehazing: Methods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 891–901, 2018.

[3] Margit Antal and László Zsolt Szabó. An evaluation of one-class and two-class classification algorithms for keystroke dynamics authentication on mobile devices. In *Control Systems and Computer Science (CSCS), 2015 20th International Conference on*, pages 343–350. IEEE, 2015.

[4] 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.

[5] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016.

[6] Paul Bodesheim, Alexander Freytag, Erik Rodner, and Joachim Denzler. Local novelty detection in multi-class recognition problems. In *2015*

IEEE Winter Conference on Applications of Computer Vision, pages 813–820. IEEE, 2015.

[7] Paul Bodesheim, Alexander Freytag, Erik Rodner, Michael Kemmler, and Joachim Denzler. Kernel null space methods for novelty detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3374–3381, 2013.

[8] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*, 2019.

[9] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[10] Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. Exploring object relation in mean teacher for cross-domain detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11457–11466, 2019.

[11] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018.

[12] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[13] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 36–51. Springer, 2017.

[14] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

[15] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[16] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3339–3348, 2018.

[17] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision*

and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 248–255. Ieee, 2009.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[20] Dipak K Dey and Jun Yan. *Extreme value modeling and risk analysis: methods and applications*. CRC Press, 2016.

[21] Akshay Raj Dhamija, Manuel Günther, and Terrance E Boult. Improving deep network robustness to unknown inputs with objectosphere.

[22] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *Proc. of ICLR*, 2017.

[23] Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134, 2016.

[24] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. A geometric framework for unsupervised anomaly detection. In *Applications of data mining in computer security*, pages 77–101. Springer, 2002.

[25] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[26] Mohammed E Fathy, Vishal M Patel, and Rama Chellappa. Face-based active authentication on mobile devices. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 1687–1691. IEEE, 2015.

[27] Raanan Fattal. Single image dehazing. *ACM transactions on graphics (TOG)*, 27(3):72, 2008.

[28] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE transactions on information forensics and security*, 8(1):136–148, 2012.

[29] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3855–3863, 2017.

[30] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[31] 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.

[32] Laurent E Ghaoui, Michael I Jordan, and Gert R Lanckriet. Robust novelty detection with single-class mpm. In *Advances in neural information processing systems*, pages 929–936, 2003.

[33] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[34] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *2011 international conference on computer vision*, pages 999–1006. IEEE, 2011.

[35] Manuel Günther, Artur Costa-Pazo, Changxing Ding, Elhocine Boutellaa, Giovani Chiachia, Honglei Zhang, Marcus de Assis Angeloni, V Štruc, Elie Khoury, Esteban Vazquez-Fernandez, et al. The 2013 face recognition evaluation in mobile environment. In *2013 International Conference on Biometrics (ICB)*, pages 1–7. IEEE, 2013.

[36] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8, 2018.

[37] Ville Hautamaki, Ismo Karkkainen, and Pasi Franti. Outlier detection using k-nearest neighbour graph. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 430–433. IEEE, 2004.

[38] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2011.

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep

into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[40] 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.

[41] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[42] Heiko Hoffmann. Kernel pca for novelty detection. *Pattern recognition*, 40(3):863–874, 2007.

[43] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[44] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Duplex generative adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1498–1507, 2018.

[45] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[46] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[47] Shehroz S Khan and Michael G Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374, 2014.

[48] Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G Macready. A robust learning approach to domain adaptive object detection. *arXiv preprint arXiv:1904.02361*, 2019.

[49] Taekyung Kim, Minki Jeong, Seunghyeon Kim, Seokeon Choi, and Changick Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12456–12465, 2019.

[50] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[51] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015.

[52] Edwin M Knorr, Raymond T Ng, and Vladimir Tucakov. Distance-based outliers: algorithms and applications. *The VLDB Journal—The International Journal on Very Large Data Bases*, 8(3-4):237–253, 2000.

[53] 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.

[54] 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.

[55] Rajesh Kumar, Vir V Phoha, and Abdul Serwadda. Continuous authentication of smartphone users by fusing typing, swiping, and phone movement patterns. In *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8. IEEE, 2016.

[56] Gert Lanckriet, Laurent E Ghaoui, Chiranjib Bhattacharyya, and Michael I Jordan. Minimax probability machine. In *Advances in neural information processing systems*, pages 801–807, 2002.

[57] Wallace Lawson, Esube Bekele, and Keith Sullivan. Finding anomalies with generative adversarial networks for a patrolbot. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 12–13, 2017.

[58] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[59] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. 2010.

[60] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 28(1):492–505, 2019.

[61] Siyuan Li, Wenqi Ren, Jiawan Zhang, Jinke Yu, and Xiaojie Guo. Single image rain removal via a deep decomposition–composition network. *Computer Vision and Image Understanding*, 2019.

[62] Y Li, R T Tan, X Guo, J Lu, and M S Brown. Rain streak removal using layer priors. *In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2736–2744, 2016.

[63] Yu Li, Shaodi You, Michael S Brown, and Robby T Tan. Haze visibility enhancement: A survey and quantitative benchmarking. *Computer Vision and Image Understanding*, 165:1–16, 2017.

[64] 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.

[65] Juncheng Liu, Zhouhui Lian, Yi Wang, and Jianguo Xiao. Incremental kernel null space discriminant analysis for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 792–800, 2017.

[66] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2):261–318, 2020.

[67] 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.

[68] Xiaohong Liu, Yongrui Ma, Zhihao Shi, and Jun Chen. Griddehazenet: Attention-based multi-scale network for image dehazing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7314–7323, 2019.

[69] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[70] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[71] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2208–2217. JMLR. org, 2017.

[72] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[73] Upal Mahbub, Sayantan Sarkar, Vishal M Patel, and Rama Chellappa. Active user authentication for smartphones: A challenge data set and benchmark results. In *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*, pages 1–8. IEEE, 2016.

[74] Markos Markou and Sameer Singh. Novelty detection: a review—part 1: statistical approaches. *Signal processing*, 83(12):2481–2497, 2003.

[75] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[76] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[77] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. *arXiv preprint arXiv:1902.00146*, 2019.

[78] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4500–4509, 2018.

[79] Hajime Nada, Vishwanath A Sindagi, He Zhang, and Vishal M Patel. Pushing the limits of unconstrained face detection: a challenge dataset and baseline results. *arXiv preprint arXiv:1804.10275*, 2018.

[80] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-100). 1996.

[81] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[82] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

[83] Poojan Oza, Hien Van Nguyen Nguyen, and Vishal M Patel. Multiple class novelty detection under data distribution shift. In *European Conference on Computer Vision*. Springer, 2020.

[84] Poojan Oza and Vishal M Patel. Active authentication using an autoencoder regularized cnn-based one-class classifier. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8. IEEE, 2019.

[85] Poojan Oza and Vishal M Patel. One-class convolutional neural network. *IEEE Signal Processing Letters*, 26(2):277–281, 2019.

[86] Poojan Oza, Vishwanath A Sindagi, Vibashan VS, and Vishal M Patel. Unsupervised domain adaption of object detectors: A survey. *arXiv preprint arXiv:2105.13502*, 2021.

[87] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 754–763, 2017.

[88] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[89] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[90] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello. Continuous user authentication on mobile devices: Recent progress and remaining challenges. *IEEE Signal Processing Magazine*, 33(4):49–61, July 2016.

[91] Vishal M Patel, Rama Chellappa, Deepak Chandra, and Brandon Barbello. Continuous user authentication on mobile devices: Recent

progress and remaining challenges. *IEEE Signal Processing Magazine*, 33(4):49–61, 2016.

[92] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine*, 32(3):53–69, 2015.

[93] Pramuditha Perera, Poojan Oza, and Vishal M Patel. One-class classification: A survey. *arXiv preprint arXiv:2101.03064*, 2021.

[94] Pramuditha Perera and Vishal M Patel. Extreme value analysis for mobile active user authentication. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 346–353. IEEE, 2017.

[95] Pramuditha Perera and Vishal M Patel. Dual-minimax probability machines for one-class mobile active authentication. In *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*. IEEE, 2018.

[96] Pramuditha Perera and Vishal M Patel. Learning deep features for one-class classification. *arXiv preprint arXiv:1801.05365*, 2018.

[97] Pramuditha Perera and Vishal M Patel. Deep transfer learning for multiple class novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11544–11552, 2019.

[98] Pedro O Pinheiro. Unsupervised domain adaptation with similarity learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8004–8013, 2018.

[99] Maarten G Poirot, Praneeth Vepakomma, Ken Chang, Jayashree Kalpathy-Cramer, Rajiv Gupta, and Ramesh Raskar. Split learning for collaborative deep learning in healthcare. *arXiv preprint arXiv:1912.12115*, 2019.

[100] Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo Regazzoni, and Nicu Sebe. Abnormal event detection in videos using generative adversarial nets. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 1577–1581. IEEE, 2017.

[101] 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.

[102] 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.

[103] Aruni RoyChowdhury, Prithvijit Chakrabarty, Ashish Singh, SouYoung Jin, Huaizu Jiang, Liangliang Cao, and Erik Learned-Miller. Automatic adaptation of object detectors to new domains using self-training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 780–790, 2019.

[104] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, Zahra Moayed, and Reinhard Klette. Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*, 2018.

[105] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018.

[106] 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.

[107] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. *CoRR*, abs/1812.04798, 2018.

[108] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

[109] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126:973–992, 2018.

[110] Babak Saleh, Ali Farhadi, and Ahmed Elgammal. Object-centric anomaly detection by attribute-based reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 787–794, 2013.

[111] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett,

editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc., 2016.

[112] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018.

[113] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[114] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[115] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *Advances in neural information processing systems*, pages 582–588, 2000.

[116] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *arXiv preprint arXiv:1911.08265*, 2019.

[117] Alexander Schultheiss, Christoph Käding, Alexander Freytag, and Joachim Denzler. Finding the unknown: Novelty detection with extreme value signatures of deep neural activations. In *German Conference on Pattern Recognition*, pages 226–238. Springer, 2017.

[118] Mohamed El Amine Seddik, Cosme Louart, Mohamed Tamaazousti, and Romain Couillet. Random matrix theory proves that deep learning representations of gan-data behave as gaussian mixtures. *arXiv preprint arXiv:2001.08370*, 2020.

[119] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[120] Abdul Serwadda, Vir V Phoha, and Zibo Wang. Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8. IEEE, 2013.

[121] Yuhu Shan, Wen Feng Lu, and Chee Meng Chew. Pixel and feature level based domain adaption for object detection in autonomous driving. 09 2018.

[122] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.

[123] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

[124] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

[125] Chandra Thapa, Mahawaga Arachchige Pathum Chamikara, and Seyit Camtepe. Splitfed: When federated learning meets split learning. *arXiv preprint arXiv:2004.12088*, 2020.

[126] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9627–9636, 2019.

[127] Philip Tresadern, Chris McCool, Norman Poh, Pavel Matejka, Abdenour Hadid, Christophe Levy, Tim Cootes, and Sebastien Marcel. Mobile biometrics (mobio): Joint face and voice verification for a mobile platform. *IEEE pervasive computing*, 2012.

[128] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[129] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[130] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[131] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018.

[132] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high

quality real rain dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12270–12279, 2019.

[133] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2008.

[134] Ted Xiao, Eric Jang, Dmitry Kalashnikov, Sergey Levine, Julian Ibarz, Karol Hausman, and Alexander Herzog. Thinking while moving: Deep reinforcement learning with concurrent control. In *International Conference on Learning Representations*, 2019.

[135] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5525–5533, 2016.

[136] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Universal domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2720–2729, 2019.

[137] Shaodi You, Robby T Tan, Rei Kawakami, Yasuhiro Mukaigawa, and Katsushi Ikeuchi. Adherent raindrop modeling, detectionand removal in video. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1721–1733, 2015.

[138] He Zhang and Vishal M Patel. Sparse representation-based open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1690–1696, 2016.

[139] He Zhang and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *arXiv preprint arXiv:1701.05957*, 2017.

[140] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2018.

[141] H. Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, abs/1802.07412, 2018.

[142] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Multi-scale single image dehazing using perceptual pyramid deep network. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 902–911, 2018.

[143] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Joint transmission map estimation and dehazing using deep networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[144] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[145] 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*, 30(11):3212–3232, 2019.

[146] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining, pages 665–674. ACM, 2017.

[147] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

[148] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 687–696, 2019.

[149] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.