DEEP LEARNING BASED NOVELTY DETECTION

by
Pramuditha Hemanga Perera

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

Baltimore, Maryland
May, 2020

© 2020 P.H. Perera
All rights reserved

Abstract

In recent years, intelligent systems powered by artificial intelligence and computer vision that perform visual recognition have gained much attention. These systems observe instances and labels of known object classes during training and learn association patterns that can be used during inference. A practical visual recognition system should first determine whether an observed instance is from a known class. If it is from a known class, then the identity of the instance is queried through classification. The former process is commonly known as novelty detection (or novel class detection) in the literature. Given a set of image instances from known classes, the goal of novelty detection is to determine whether an observed image during inference belongs to one of the known classes.

In this thesis, deep learning-based approaches to solve novelty detection is studied under four different settings. In the first two settings, availability of out-of-distributional data (OOD) is assumed. With this assumption, novelty detection can be studied for cases where there are multiple known classes and a single known class separately. These two problem settings are referred to as *Multi-class novelty detection with OOD data* and *one-class novelty detection with OOD data* in the literature, respectively. It is also possible to study this problem in a more constrained setting where only the data from known classes are considered for training. When there exists multiple classes in this setting novelty detection problem is known as *Multiple-class novelty detection or Open-set recognition.* On the other hand, when only a single class exists it is known as *one-class novelty detection.*

Finally, we study a practical application of novelty detection in mobile Active Authentication (AA). For a practical AA-based novelty detector, latency and efficiency are as important as the detection accuracy. Solutions are presented for the problem of quickly detecting intrusions with lower false detection rates in mobile AA systems with higher resource efficiency. Bayesian and Minimax versions of the Quickest Change Detection (QCD) algorithms are introduced to quickly detect intrusions in mobile AA systems. These algorithms are extended with an update rule to facilitate low frequency sensing which leads to low utilization of resources.

Thesis Readers

Dr. Vishal Patel (Primary Advisor)
Assistant Professor
Department of Electrical and Computer Engineering
Johns Hopkins University

Prof. Trac D. Tran
Professor
Department of Electrical and Computer Engineering
Johns Hopkins University

Dedicated to my Father Ranil, Mother Saumya and Wife Hasara
for all the love and support granted over years.

Acknowledgements

Working towards a doctoral degree over a period of five years took a lot of effort and dedication. I would not have made it without the support and encouragement of many individuals around me.

First and foremost I would like to thank my adviser Vishal for guiding me through this journey. I consider myself lucky to have an adviser who is friendly and approachable as Vishal is. I find his style of advising inspiring and surprisingly effective. He has been a very thoughtful and understanding when I had to go through various milestones in my personal life. Thank you Vishal for all the support and guidance given through the years.

I wish to express my gratitude to Prof. Alan Yuille, Prof.Gregory Hager and Dr.Mark Dredze who officiated in my Graduate Board Examination panel. I would like to pay a special thanks to Prof. Trac Tran and Dr. Najim Dehak for officiating in my dissertation committee, acting as dissertation readers, and for giving me valuable feedback throughout the process.

During my study, I had the privilege to collaborate in research with Amazon and Adobe research teams. I find the experience I gained there to be invaluable. I would like to thank Dr. Vlad Morariu, Dr. Rajiv Jain, Dr. Bing Xiang, and Dr. Ramesh Nallapati for their support and motivation during our collaborations.

I wish to show my gratitude to all teachers in Kingswood College, Sri Lanka and lecturers at the University of Peradeniya, Sri Lanka. Especially, I would like to pay my special regards to my undergraduate advisors Dr. Janaka Wijayakulasooriya, Dr.Roshan Godaliyadda and Dr. Parakrama Ekanayake who were instrumental in making me interested in academic research. I wouldn't have persuaded a doctoral program if not for their motivation and encouragement.

Finally, I would like to thank my family for their continued support after all these years. I would like to remember my Farther Ranil who is no longer with us. Especially, I would like to thank my mother Saumya and wife Hasara who have been the two pillars of my life over the years.

Contents

Abstract ii

Dedication iv

Acknowledgements v

Contents vii

List of Figures xii

Chapter 1 Introduction 1

Related Work 5

One-class Novelty Detection 5

Open-set Recognition 7

Multi-class Novelty Detection 8

Adversarial Learning 9

Self-Supervision 10

Chapter 2 Background 11

Deep Classification Networks 11

Cross-entropy Loss 12

Autoencoder Networks 12

Generative Adversarial Networks (GANs) 13
One-class Classification 14
    One-class Support Vector Machines 14
    Support Vector Data Descriptor(SVDD) 15

Chapter 3 Multiple-class Novelty Detection with OOD Data 17
Positive Filters 18
Proposed Method 20
    Limitations of Cross-Entropy Loss 21
    Membership Loss 22
    Globally Negative Filters 23
    Training Procedure 25
    Testing (Novelty Detection) 26
Experimental Setup and Results 27
    Baseline Methods 28
    Datasets 29
    Results 30
    Ablation Study 32
    Impact of the Reference Dataset 33
    Impact on Classification Accuracy 34
    Network Hyper-parameters 34
    ROC Curves 35
Summary 35

Chapter 4 One-class Novelty Detection with OOD Data 38

Objective Function 41
Proposed Loss Functions 45
Terminology 46
Architecture 48
Compactness loss 48
Training 49
Testing 51
Memory Efficient Implementation 52
Experimental Results 53
Experimental Setup 53
Results 55
Discussion 61
Impact of Different Features 63
Impact of the Reference Dataset 64
Derivations 65
Detailed Network Architectures 67
Summary 67

Chapter 5 One-class Novelty Detection **69**
Motivation 71
Proposed Strategy 72
Experimental Results 80
Evaluation Methodology 80
Datasets and Experimental Results 80
Ablation Study 83

Summary 84

Chapter 6 Multiple-class Novelty Detection/ Open-set Recognition 85

Proposed Method 87

Challenges in Open-set Recognition 87

Self-Supervised Learning 88

Augmenting with Generative Representation 89

Training and Testing Procedure 90

Experimental Results 92

Open-set Recognition 93

Out-of-distributional Detection 95

Case Study and Ablation Study 96

Summary 101

Chapter 7 Data Efficient Novelty Detection with Low Latency . . 102

Intruder Detection in AA 106

Average Detection Delay (ADD) 108

Probability of False Detections (PFD) 108

Quickest Change Detection 110

Bayesian QCD (BQCD) 110

MiniMax QCD (MQCD) 112

CumSum Algorithm 113

Proposed Algorithm 113

Resource Efficient Quickest Change Detection 116

Efficient Bayesian Formulation (E-BQCD) 117

Efficient MiniMax Formulation (E-MQCD) 118

Modified Algorithm 119

Experimental Results 122

Quickest Change Detection 122

Resource Efficient QCD 128

Discussion 133

Summary 133

Chapter 8 Discussion and Future Work **135**

References **138**

Curriculum vitae **145**

List of Figures

Figure 1-1 Different formulations of the Novelty Detection problem. . . 2

Figure 2-1 Types of deep learning networks. 12

Figure 2-2 Different formulations of the one-class classification problem. 15

Figure 3-1 Positive and negative filters of the *sand snake* class in the Resnet50 trained on ILSVRC12 dataset. 19

Figure 3-2 (a) Activations of known (Calculator) and unknown samples (Playing Cards) in a VGG16 model. 21

Figure 3-3 Proposed architecture for multiple-class novelty detection with OOD data. 27

Figure 3-4 Sample images from multiple-class novelty detection evaluation datasets. 29

Figure 3-5 ROC curves obtained for novelty detection in the Caltech256 dataset. 35

Figure 3-6 ROC curves obtained for novelty detection in the CUB200 dataset. 35

Figure 3-7 ROC curves obtained for novelty detection in the Founder-Type200 dataset. 36

Figure 3-8 ROC curves obtained for novelty detection in the Stanford Dogs dataset. 36

Figure 4-1 Different deep learning strategies used for classification. . . . 40

Figure 4-2 Overview of the proposed method. 41

Figure 4-3 Possible strategies for one-class classification in abnormal image detection. (a)Image samples. (b) AlexNet features. (c) Binary CNN (d) Fine-tuning (e) Proposed method. 43

Figure 4-4 (a) Training, and (b) testing frameworks of the proposed DOC method. 47

Figure 4-5 Variation of loss functions during training in the proposed method. 51

Figure 4-6 Sample images from datasets used for evaluation. (a)PASCAL VOC + Abnormal 1001 dataset. (b) Caltech 256 dataset. (c) UMDAA02 dataset. 55

Figure 4-7 Sample false detections for the one-class problem of novelty detection (*American Flag*). 61

Figure 4-8 CNN architectures based on AlexNet and VGG16 backbones for the proposed method. 68

Figure 5-1 Limitations of in-class representation based novelty detection. 70

Figure 5-2 Illustration of the latent space learned for digit 8 using a denoising-autoencoder network (left) and OCGAN(right). . . 72

Figure 5-3 Visualization of generated images from random latent samples when the network is trained (a) without and (b) with informative-negative mining. 75

Figure 5-4 Effectiveness of informative-negative mining. 77

Figure 5-5 Illustration of OCGAN architecture: the network consists of four sub-networks : an auto-encoder, two discriminators and a classifier. 79

Figure 5-6 Representative images from the datasets used for evaluation. Images in each column belong to the same class. 81

Figure 6-1 (a) A classifier defines a positive half space for each class. (b) An open-set object could project either near a decision boundary (samples B and C) or deep into the positive half space (samples A and D) of a given class. (c) We learn a classifier which takes into account more factors than just class separation. 86

Figure 6-2 Comparison of the network architectures. (a) A CNN network. (b) CNN with self-supervision. (c) Proposed network. 88

Figure 6-3 Score histograms for open-set and known-set samples. . . . 97

Figure 6-4 tSNE visualization of the feature space for (a)Conventional CNN and for the (b) proposed method. 98

Figure 6-5 Top Row: Visualization of open-set samples. Middle Row: z-score normalized activations produced by the baseline CNN. Bottom Row: z-score normalized activations generated by proposed method. 99

Figure 6-6 Reconstructed images produced by the auto encoder trained on known-set images. 100

Figure 7-1 The problem of quick intrusion detection in face-based AA systems. (A-I) show the genuine user with varying facial expressions. An intrusion occurs starting from (J). 104

Figure 7-2 An overview of the proposed QCD-based AA method. 104

Figure 7-3 An overview of a typical AA system. 107

Figure 7-4 Sample PFD-ADD curves of two AA systems. 109

Figure 7-5 Variation of Bayesian QCD scores for the video shown in Figure 7-1. 116

Figure 7-6 Variations of efficient Bayesian QCD scores for the video shown in Figure 7-1 for different APO values. 121

Figure 7-7 Sample detected face images from (a) the MOBIO dataset and (b) the UMDAA-01 dataset. 123

Figure 7-8 Performance curves obtained on the UMDAA-01 face dataset. 124

Figure 7-9 Performance curves obtained on the UMDAA-01 touch dataset. 125

Figure 7-10 Performance curves obtained on the MOBIO face dataset. . . 126

Figure 7-11 Performance curves obtained on Touchalytics dataset. 127

Figure 7-12 Performance curves obtained on the UMDAA-01 face dataset for efficient QCD. 129

Figure 7-13 Performance curves obtained on the UMDAA-01 touch dataset for efficient QCD. 129

Figure 7-14 Performance curves obtained on the MOBIO face dataset for efficient QCD. 130

Figure 7-15 Performance curves obtained on the Touchalytics touch dataset for efficient QCD. 130

Figure 7-16 Effect of using different APO values for sampling. 132

Figure 7-17 Effect of sampling on minimum average detection delay. . . . 132

Chapter 1

Introduction

Supervised classification systems are trained with the knowledge of a finite set of labeled training examples. When training data comes from c distinct known classes, a classifier simultaneously learns a descriptive feature space and a decision rule that segments the feature space into c non-overlapping regions. When an object outside the known class set (known as a novel object) is introduced to the network, the network will still associate it with one of the known c classes. The goal of novelty detection is to identify whether a given object instance belongs to the known class set or not. Once identified, open-set samples can be either discarded to prevent wrong association or used to improve the classification system [1].

The role of a novelty detector in a classification framework is illustrated in Figure 1-1(Top-left). Consider a wild animal classifier trained on three classes {Bear, Camel, Elephant }. These three classes are known to the classifier (hence $c = 3$). Objective of the novelty detector is to separate objects belonging to any other novel class from known classes. For example, images of a fish, frog, dog and duck all will be considered to be novel samples in this context since they were not included in the known class set. Novelty detector essentially defines a *novelty boundary* that separates known class samples from the rest of the world.

Novelty detection is encountered in many real-world computer vision applications

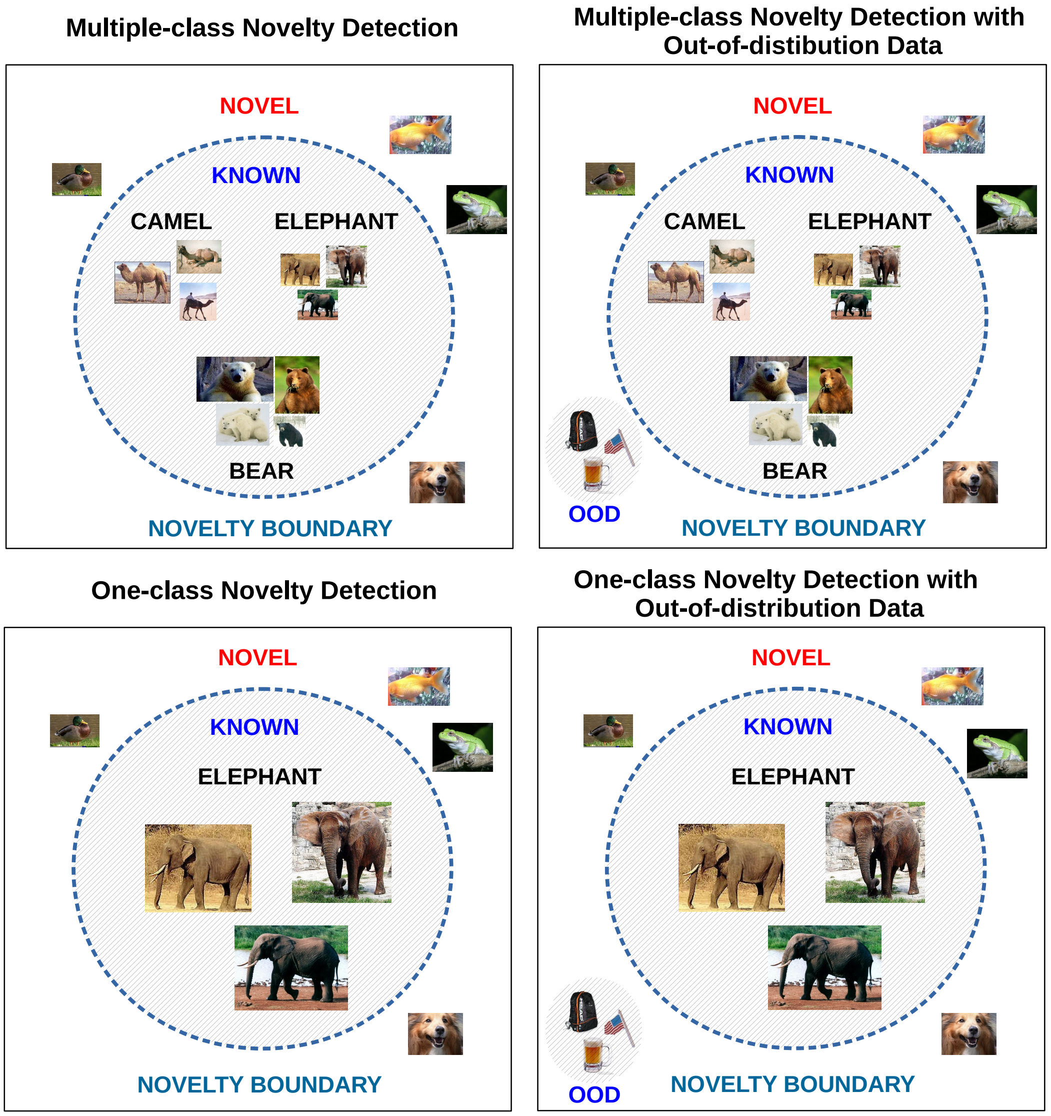

Multiple-class Novelty Detection
Multiple-class Novelty Detection with Out-of-distibution Data
NOVEL
KNOWN
CAMEL
ELEPHANT
BEAR
NOVELTY BOUNDARY
NOVEL
KNOWN
CAMEL
ELEPHANT
BEAR
OOD
NOVELTY BOUNDARY
One-class Novelty Detection
One-class Novelty Detection with Out-of-distribution Data
NOVEL
KNOWN
ELEPHANT
NOVELTY BOUNDARY
NOVEL
KNOWN
ELEPHANT
OOD
NOVELTY BOUNDARY

Figure 1-1. Different formulations of the Novelty Detection problem.

including outlier detection [2], anomaly detection [3], [4], medical imaging and mobile active authentication [5],[6],[7]. In all of these applications, unavailability of samples from novel classes is either due to the openness of the problem or due to the high cost associated with obtaining the samples of such classes. For example, in an outlier detection application, it is counter intuitive to come up with outlier samples to train a classifier. On the other hand, in mobile active authentication, samples of alternative classes (users) are often difficult to obtain due to the privacy concerns [8]. Depending on the resources available for training, novelty detection problem can be studied under four different settings. These four settings are illustrated in Figure 1-1.

Multiple-class Novelty Detection. Identify whether a given object instance belongs to the known class set or not. Only known class objects may be used during the training process. This problem is often referred to as the open-set classification problem in the literature [9]. This setting is illustrated in Figure 1-1(Top-left).

Multiple-class Novelty Detection with Out-of-distribution Data. Given a set of image instances from known classes and a set of out-of-distributional data, the goal of novelty detection is to determine whether an observed image during inference belongs to one of the known classes as shown in Figure 1-1(Top-right). Out-of-distributional data (OOD) are multiple-class annotated data from a different problem domain [10]. For example, for a wild animal classifier, images of backpacks, flags and mugs can be considered to be OOD. In this setting, deep networks trained on out-of-distributional data may be used during training. This setting is simply referred to as Multiple-class Novelty Detection in the literature [11].

One-class Novelty Detection. One-class novelty detection tackles the problem of quantifying the probability that a test example belongs to the distribution defined by training examples [12]. In one-class novelty detection, examples of only a single class are observed during training. Therefore, deep features or deep models cannot be utilized in this setting. One-class novelty detection is an extreme version of multiple-class novelty detection as shown in Figure 1-1(bottom-left). The only difference in this setting is that there exists only a single class as opposed to the former. Standard one-class classification solutions can be applied to solve this problem.

One-class Novelty Detection with Out-of-distribution Data. Objective of this setting is to recognize instances of a concept by using examples of the same concept when annotated out-of-distributional data is available during training. As a result, out-of-distributional data and deep models trained on such data may be used during training. This setting is illustrated in Figure 1-1(bottom-right).

Out of all these four settings, *Multiple-class novelty detection with out-of-distribution data* is the easiest setting to perform novelty detection on. With multiple known classes along with annotated out-of-distributional data available, it is possible to learn a highly informative embedding that will allow novelty detection more effective. Since multiple class data of some form (either in known class set or in the out-of-distributional set) are available, it is possible to learn a deep network-based representation in *open-set detection* and *one-class transfer learning* settings as well. On the other hand, *one-class novelty detection*, where training is permitted to use only samples from a single known class cannot exploit pre-trained networks or features. Thus, it becomes a highly challenging problem.

Related Work

One-class Novelty Detection

One-class novelty detection is a well-defined research problem with standard evaluation metrics that has received considerable attention in the recent past. It has been traditionally treated as a representation learning problem. The earliest methods in one-class novelty detection used Principal Component Analysis (PCA) [13] and its kernel extension [14] to find a subspace that best describes the given concept. With the advent of neural networks and deep learning, a similar mapping was sought using auto-encoder networks [15]. Once such a mapping is learned, one-class novelty detection is carried out either based on reconstruction error or by explicitly modeling the normal behaviour of the known class in the latent space. In [14] and [16] the former strategy has been used to perform novelty detection using mean squared error as the novelty function. In [17], a Generative Adversarial Network (GAN) [18] is trained to de-noise noisy samples of the given class. There, the discriminator's prediction in the image space is used to quantify reconstruction error. Following a slightly different strategy, [19] proposes to learn a mapping between a random distribution and the image manifold of the given class. In [19], the closest image to a query is sought through back-propagation, where novelty detection is performed based on the difference between the two images.

The latter strategy, where the behavior of the known class in the latent space is modeled, has also received considerable attention in recent works. Earlier work of this nature used one-class modeling tools such as One-class SVM [20] and Support Vector Data Descriptor (SVDD) [21] on top of an obtained latent representation. One class Support Vector Machines (SVMs) treats the origin as the out-of-class region and tries to construct a hyperplane separating the origin with the class data.

Using a similar motivation, [21] proposed Support Vector Data Description (SVDD) algorithm which isolates the training data by constructing a spherical separation plane. In [22], first, a GAN is used to obtain a latent representation. Then, the probability distribution of the latent space is modeled as a product of two marginal distributions where marginal distributions are learned empirically. In contrast, in [23] the latent distribution is modeled using an auto-regressive network that is learned along with the parameters of the auto-encoder. Using a different approach, deep-SVDD [12] tries to learn a latent space where intra-class variance is low. The method proposed by [12] is conceptually similar to [24] but does not use any external data in finding the solution as done in the latter work.

In [25], visual anomalies in wire ropes are detected based on Gaussian process modeling. Anomaly detection is performed by maximizing the KL-divergence in [26], where the underlying distribution is assumed to be a known Gaussian. A detailed description of various anomaly detection methods can be found in [3]. Some of the earlier works in novelty detection focused on estimating a parametric model for data and to model the tail of the distribution to improve classification accuracy [27],[4]. In [28], null space-based novelty detection framework for scenarios when a single and multiple classes are present is proposed. However, it is mentioned in [28] that their method does not yield superior results compared with the classical one-class classification methods when only a single class is present. An alternative null space-based approach based on kernel Fisher discriminant was proposed in [29] specifically targeting one-class novelty detection. A detailed survey of different novelty detection schemes can be found in [2], [30].

Mobile-based Active Authentication (AA) is another application of one-class learning which has gained interest of the research community in recent years [5]. In mobile AA, the objective is to continuously monitor the identity of the user based on his/her enrolled data. As a result, only the enrolled data (i.e. one-class data) are available during training. Some of the recent works in AA has taken advantage of

CNNs for classification. Work in [31], uses a CNN to extract attributes from face images extracted from the mobile camera to determine the identity of the user. Various deep feature-based AA methods have also been proposed as benchmarks in [8] for performance comparison.

Since one-class learning is constrained with training data from only a single class, it is impossible to adopt a CNN architectures used for classification [32], [33] and verification [34] directly for this problem. In the absence of a discriminative feature generation method, in most unsupervised tasks, the activation of a deep layer is used as the feature for classification. This approach is seen to generate satisfactory results in most applications [35]. This can be used as a starting point for one-class classification as well. As an alternative, autoencoders [36], [15] and variants of autoencoders [37], [38] can also to be used as feature extractors for one-class learning problems. However, in this approach, knowledge about the outside world is not taken into account during the representation learning. Furthermore, none of these approaches were specifically designed for the purpose of one-class learning.

Open-set Recognition

Open-set recognition has received considerable attention in the computer vision community in recent years. The problem of open-set recognition was first formulated in [9], where authors pointed out the possibility of an open-set sample generating a very high activation score for one of the known class categories. Since then, several other works have analyzed this challenge in the context of deep networks [39],[40]. In [41], a $k+1$ classifier for a k class problem was used where the extra class was treated as the *open-set class*. A statistical method was used to apportion class probabilities to the open-set class. This alternative formulation, OpenMax, was proposed as an alternative to the SoftMax operator. In [42], a Generative Adversarial Network (GAN) based framework was used to estimate open-set class activations. A similar approach was taken in [43] where counterfactual images that lie between decision boundaries were used to simulate open-set class instances.

More recent works in open-set recognition have deviated from simulating open-set classes. The method proposed in [44] used a class conditioned generator to learn a representation that preserves only known-class samples. Then, open-set recognition was carried out based on the reconstruction error associated with the generator. In [45], the authors identified the importance of generative features in open-set recognition. They first learn a sophisticated generative model (an extension of a ladder network [46]) and append the learned feature with one of the classifier features. Then, an OpenMax classifier was learned using the augmented features. The feature augmentation proposed in our work is different from [45]. In [45], a generative model and a classifier are trained independently. We learn a classifier trained on the augmented input space and take into account the disparity between the two representations as we compute class activation scores.

Multi-class Novelty Detection

Object classification schemes are often equipped with a suitable mechanism to detect novel objects. For example, Eigenfaces [47] was accompanied by a reconstruction error-based novel object detection method. In sparse representation-based classification (SRC) algorithm [48], Sparsity Concentration Index (SCI) was proposed for the same purpose. In contrast, there is no formal novelty detection mechanism proposed for deep-learning based classification. In its absence, thresholding the highest class activation score of the deep model has been used as a baseline in the literature [49]. As an alternative, several recent works have proposed novelty detection schemes based on deep features [49],[50]. In the same spirit, it is also a possibility to use classical novelty detection tools such as Kernel PCA [14], Kernel null space-based novelty detection (KNFST) [28] and its variants [51],[11] on deep features. KNFST operating on deep-features produces the current state of the art performance in visual novelty detection [11]. However, advantages of deep-learning are not properly exploited in all of these approaches due to the absence of an end-to-end learning framework.

On the other hand, novelty detection problem has a close resemblance to both anomaly detection [52], [3],[6] and open-set recognition problems [9],[49]. Therefore, it is possible to solve anomaly detection using tools proposed in these alternative domains. In anomaly detection, given a single *normal* class, the objective is to detect out-of-class instances. One-class SVM [20] and SVDD [21] are two of the most widely used tools in anomaly detection. Novelty detection can be viewed as an anomaly detection problem if all known classes are considered as a single augmented class. On the other hand, objective in open-set recognition (OSR) is similar to that of novelty detection. But in addition, OSR requires correct classification of samples detected as known samples. Therefore, it is also possible to use open-set recognition tools to perform novelty detection. However, we note that due to subtle differences in objectives, OSR algorithms are not optimal for novelty detection.

Adversarial Learning

Given a set of images, Generative Adversarial Networks (GANs) introduced in [18] play a two-player game between a generator network and a discriminator network. Here, the generator network tries to produce realistic images (fake images) from the given image distribution whereas the discriminator network tries to distinguish fake images from real images. At equilibrium, the generator network learns the distribution of the given image set. In order to achieve this state, GAN theory dictates that there should be a balance between the capacities of the two networks. In [53], GAN was extended to the conditional setting. Based on this extension, GANs have been used in many image-to-image translation applications. It was shown in [54] that GANs can be used to learn stable representations even with deep convolutional networks, provided that certain design choices are made.

Self-Supervision

Self-supervision is an unsupervised machine learning technique where data itself provides supervision. It is usually carried out in addition to a primary objective (such as classification or detection) with the intention of producing a more generic and robust feature. Recent works in self-supervision introduced several techniques to improve the performance in classification and detection tasks. In all of these techniques, the network is forced to learn the shape structures of the underlying objects and their semantics thereby producing a richer feature.

For example, in [55], given an anchor image patch, self-supervision was carried out by asking the network to predict the relative position of a second image patch. To make such predictions, the network needs to learn object structure and relative order. In [56], a multi-task prediction framework extended this formulation, forcing the network to predict a combination of relative order and pixel color. In [57], the image was randomly rotated by a factor of 90 degrees and the network was forced to predict the angle of the transformed image. This method was simpler to implement and produced better results than previous self-supervision techniques. In our work, we follow [57] by using a series of different transformations (combination of rotating and flipping the image) in place of rotations. To the best of our knowledge this is the first attempt at using self-supervision for open-set recognition. The prediction of geometric transformations has been previously utilized in [58] in the one-class classification problem domain. However, [58] is different from our method as they used this network to generate classifier responses to characterize a signature for a given class.

Chapter 2

Background

In this chapter, we give a brief review of concepts in deep learning and one-class classification. Specifically, we provide a brief background on deep-convolutional classifier networks, deep-autoencoders, Generative Adversarial Networks (GAN), one-class Support Vector Machines (OCSVM) and Support Vector Data Descriptor (SVDD).

Deep Classification Networks

Consider a c class fully-supervised object classification problem with a training image set $\mathbf{x} = x_1, x_2, \ldots, x_n$ and the corresponding labels $\mathbf{y} = y_1, y_2, \ldots, y_n$ where $y_i \in \{1, 2, \ldots c\}$. Deep convolutional neural networks seek to learn a hierarchical, convolutional filter bank with filters that respond to visual stimuli of different levels. Typically a network consists of two sub-networks in cascade; a feature extraction network g and a classifier network h as shown in Figure 2-1(a). In c class classification, the top most convolutional filter activation $\mathbf{g}$ is subjected to a non-linear transformation through the classifier network to generate the final activation vector $\mathbf{f} \in \mathbb{R}^c$ (for example, $\mathbf{g}$ is the conv5-3 layer in VGG16 [33] and conv5c in Resnet50 [59]. $\mathbf{f}$ is the fc8 and fc1000 layers in the respective networks). In a supervised setting, network parameters are learned such that $\arg\max \mathbf{f} = y_i$ for $\forall i \in \{1, 2, \ldots, n\}$. This is conventionally done by optimizing the network parameters using the cross-entropy

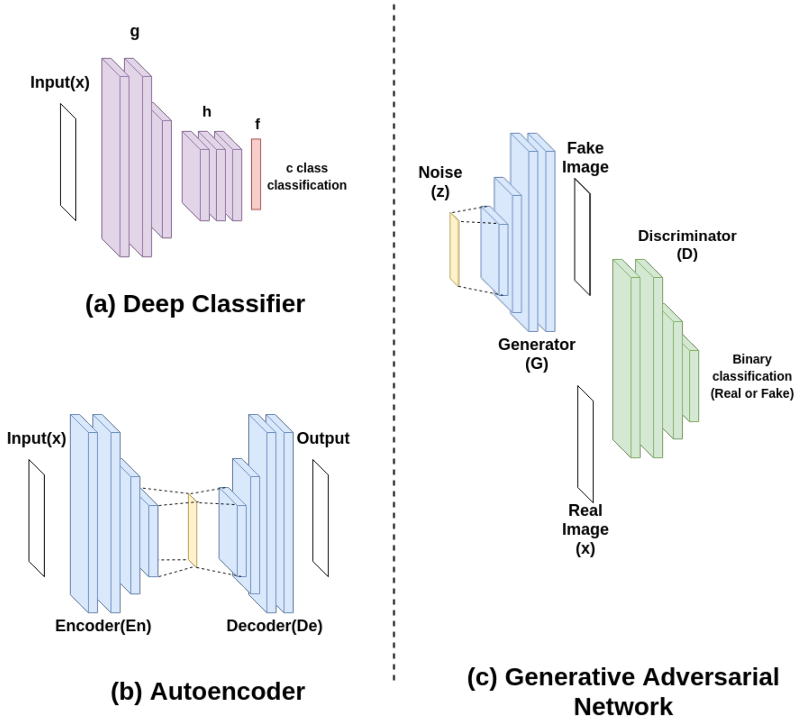


Figure 2-1. Types of deep learning networks.

loss.

Cross-entropy Loss

Cross-entropy loss is a widely used loss function in classification network training. Minimizing cross-entropy loss encourages an embedding that maximizes logit belonging to the ground truth class relative to other classes. Mathematically it is defined as follows,

$$-\log \frac{e^{f_{y_i}}}{\sum_{i=1}^{c} e^{f_{y_j}}}, \tag{2.1}$$

where y_i is the ground truth label of the input and c is the total number of classes.

Autoencoder Networks

The auto-encoder is an encoder (En) - decoder (De) structure as shown in Figure 2-1(b). It is trained with the objective of minimizing the distance between the input and the output of the network. In theory, any distance measure can be considered to learn parameters of the autoencoder. For example, mean squared error defined as,

$$l_{\text{MSE}} = \|x - \text{De}(\text{En}(x))\|_2^2, \tag{2.2}$$

where x is the input image, can be considered for this purpose. Both encoder and decoder networks consist of standard neural network components. It is the usual practice to have a bottleneck latent-space in between with a dimension smaller than the input. Due to this bottleneck, auto-encoder retains only essential information in the latent space that is required for reconstruction. It has been shown in the literature that adding noise to the input can reduce over-fitting and improve generalizabilty of the network. When noise is added to the input, the network is referred to as a *denoising autoencoder* [36]. In a denoising auto-encoder, given a noisy image, the network is expected to reconstruct the denoised version of the image. Denoising auto-encoders open up the possibility of having a latent dimension larger than the input image dimension [36].

Generative Adversarial Networks (GANs)

Generative Adversarial Network is a type of a generative model that consists of two sub-networks - a Generator (G) and a Discriminator (D). The set of images used to train a GAN is referred to as *real images.* The goal of the generator network is to use a random noise vector z to generate images that closely resemble *real images.* Images generated by the generator is referred to as *fake images.* The goal of the discriminator is to differentiate between *real images* from *fake images.*

In order to achieve this objective, discriminator is designed to generate a high score for *real images* and a low score for *fake images.* Therefore, discriminator parameters are learned such that $\log D(x)$ and $\log(1 - D(G(z)))$ are maximized, where x and z are *real image* samples and random noise vectors, respectively. Therefore, optimization in discriminator update becomes,

$$\max_D \mathbb{E}_{x\sim p_x}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))]. \tag{2.3}$$

On the other hand, the goal of the generator is to produce highly realistic *fake images* that are good enough to fool the discriminator. Therefore, parameters of G are learned such that the above loss is minimized. Hence, the full GAN training objective becomes,

$$\min_G \max_D \mathbb{E}_{x\sim p_x}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))]. \tag{2.4}$$

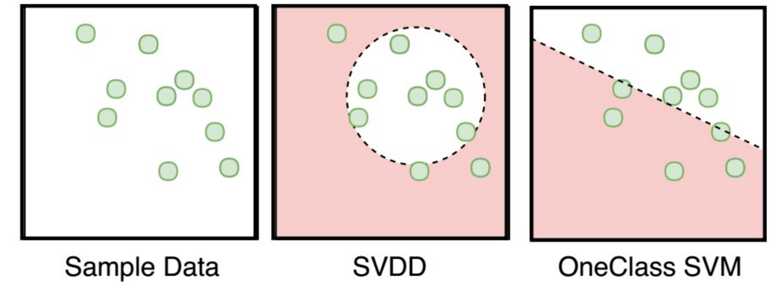
In practice, a GAN is trained by iteratively updating discriminator and generator networks. At equilibrium, generator learns to produce realistic *fake images* from random noise and the distribution of *fake images* approaches that of *real images*[18].

One-class Classification

In one-class classification, a decision boundary is defined for a set of given data in a chosen feature space. One-class Support Vector Machines(OCSVM)[20] and Support Vector Data Descriptor(SVDD)[21] are two of the widely used formulations in one-class classification.

One-class Support Vector Machines

One-class SVM is a special case of the standard SVM formulation. In the standard SVM formulation, a maximum margin decision boundary is sought that can separate positive and negative data points in a feature space. In the absence of any negative data, in one-class SVM, the origin of the coordinate system is treated as a proxy for negative data. Therefore, the optimization tries to find the hyper-plane that is furthest away from the origin that separates positive data from the origin. This hyper-plane



Sample Data SVDD OneClass SVM

Figure 2-2. Different formulations of the one-class classification problem.

can be found by solving the following optimization problem for w and ρ:

$$\begin{aligned} \min_{w,\xi,\rho} \quad & \frac{1}{2}||w||^2 + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i - \rho \\ \text{s.t.} \quad & w\phi((x_i) > \rho - \xi_i \\ & \xi \geq 0, \end{aligned} \tag{2.5}$$

where x_i, $\xi_i \forall i$ are data points and slack variables, respectively. $\phi(.)$ is the feature extraction function used to extract features. During inference, if a test sample x_t satisfies $w\phi((x_i) > \rho$ then it is declared as an instance of the positive class. In Figure 2-2(left), a set of sample data is illustrated in a 2D feature space. In Figure 2-2(right), obtained one-class SVM classification boundary is illustrated. Negative half space defined by this boundary is shaded in red.

Support Vector Data Descriptor(SVDD)

In SVDD, the hyper-sphere with the lowest radii that can encapsulate training data is sought through an optimization procedure. Specifically, the following optimization problem is solved to find R and a:

$$\begin{aligned} \min_{R,a} \quad & R^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \|x_i - a\|^2 < R^2 + \xi_i \\ & \xi \geq 0. \end{aligned} \tag{2.6}$$

Learned parameters R and a are interpreted as the radii and the center of the hyper-sphere. During inference, if a test sample x_t satisfies $\|x_t - a\|^2 < R^2$, then it is declared to be from the positive class. In Figure 2-2(middle), the decision boundary obtained by SVDD for a given sample data is illustrated.

Chapter 3

Multiple-class Novelty Detection with OOD Data

Given a set of known classes from a certain problem domain, generally unknown class data from the same problem domain is unavailable. However, in some cases it is possible to obtain data outside the known class from different problem domains, which are referred to as *out-of-distributional* samples. For example, for a face recognition application, ImageNet dataset [60] that contains images of objects may be considered as *out-of-distributional* samples. However, since the *out-of-distributional* data are from a different problem domain, they do not approximate the distribution of the *novel* samples well.

Nevertheless, since the deep-models produce generalizable features, the knowledge of *out-of-distributional* samples can be transferred to the original problem to aid novelty detection. When the problem considered is a c class problem, and when the *out-of-distributional* data of $\mathcal{C}$ classes are available, the following three strategies are used to transfer the knowledge for novelty detection in the literature:

1. **Fine-tuning**: Network is first pre-trained on the *out-of-distributional* data and later fine-tuned on the training data of the given domain. Novelty is queried by thresholding the final activation score [49].

2. **Feature Extraction**: Conventional novelty detection techniques [28],[11],[50] are

used based on the fine-tuned features.

3. **Fine-tune** $(c + \mathcal{C})$: Network is first pre-trained on the *out-of-distributional* data. Both the training data and the *out-of-distributional* data are used to perform fine-tuning in $(c + \mathcal{C})$ classes together. Novelty is determined in the same way as in approach 1.

We note that in all these baselines, the *out-of-distributional* data is employed in the training process. In fact, any novelty detection method operating on the pre-trained/finetuned deep features are implicitly making use of the *out-of-distributional* data. In this chapter, we introduce a new framework to perform novelty detection based on transfer learning. First, we show that using cross-entropy loss alone for training is not optimal for the novelty detection task. Secondly, we empirically show that the *out-of-distributional* data can be used more effectively in training to produce better novelty detection performance with respect to considered baseline solutions. Specifically, we make following primary contributions in this paper.

Positive Filters

In section 2 a brief introduction to deep-classification networks was provided. Consider a classification network trained on c classes with a convolutional feature extractor network g. If there exist k filters in the top most convolution filter bank, its output $\mathbf{g}$ is a set of k number of activation maps. The final activation vector of the network $\mathbf{f}$ is a function of $\mathbf{g}$. For a given class i, there exists some k_i filters in the filter bank $(1 \leq k_i \leq k)$ that generally generates positive activation values. These activations provide supporting (positive) evidence that an observed image is from class i. Conversely, all remaining filters provide evidence against this hypothesis. Activation score of each class in $\mathbf{f}$ is determined by taking into account the evidence for and against each class. For the remainder of the paper, we call filters that provide evidence for and against a

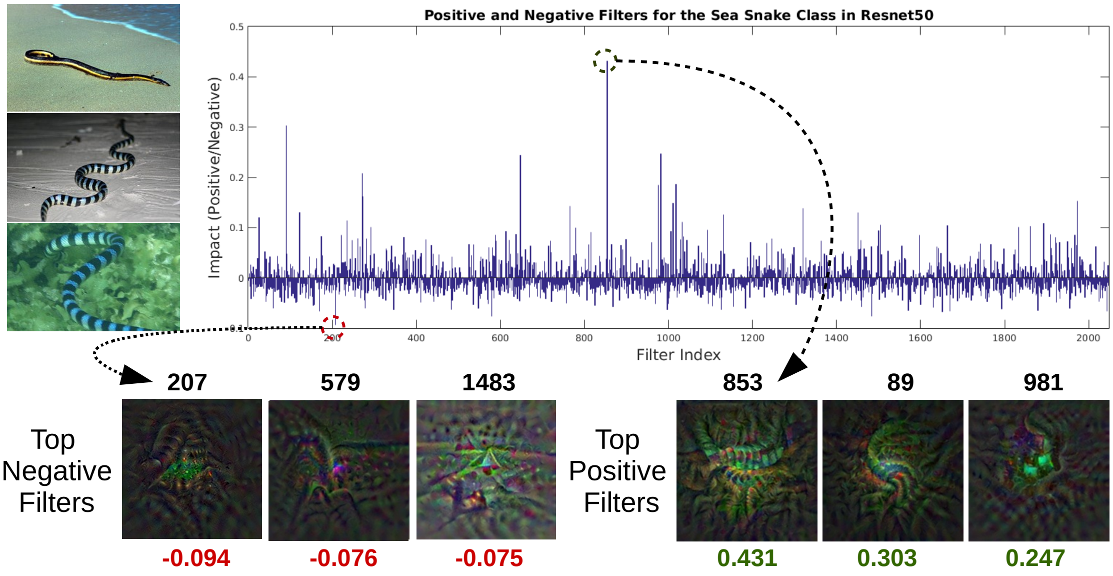


Figure 3-1. Positive and negative filters of the *sand snake* class in the Resnet50 trained on ILSVRC12 dataset.

particular class as *positive filters* and *negative filters* of the class, respectively.

This concept can be easily explained by taking the Resnet architecture [59] as an example. In Resnet, final convolution output **g** is subjected to global average pooling followed by a fully connected layer. Therefore, the i^{th} component of the final activation vector **f** can be written as $\mathbf{f}_i = W_i \times GAP(g)$, where GAP is global average pooling operation (mean of filter map) and W is the weight matrix of the fully connected layer. Here, activation of the i^{th} class is a weighted summation of mean feature maps found in **g**. From the above definition, filters associated with positive weights for a given class in W can be identified as *positive filters* for that particular class. Conversely, filters associated with the negative weights become *negative filters* of the class.

For example consider the *Sand Snake* class appearing in the ILSVRC12 dataset [60]. Shown in Figure 3-1 (top) are the weights associated with the *Sand Snake* class in the final fully connected layer of the Resnet50 network trained on the ILSVRC12 dataset. We recognize filters associated with positive and negative weights as positive and negative filters, respectively for the given class. In Figure 3-1 (bottom) we visualize per-unit visualization of top positive and top negative filters for the considered class using the DeepVis toolbox [61] (these are the images that are most likely to activate the corresponding filters). By observation, we notice that the top *positive filters* are activated when the network observes structures similar to snakes. On the other hand, the top *negative filters* are unrelated to the appearance of *sand snakes*.

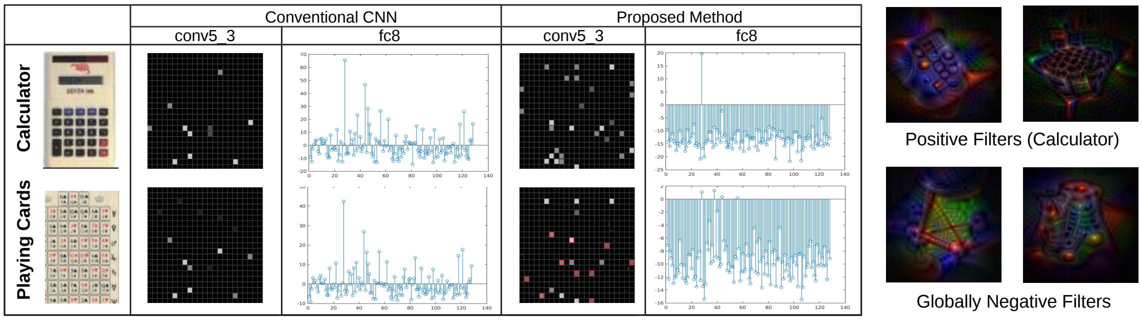
Proposed Method

Based on the above background, we propose to learn the distributions of known object classes using a CNN framework with the objective of performing joint classification and novelty detection. In our formulation, assuming each known class has a unique single label, we force the final activation vector $\mathbf{f}$ to model the probability distribution vector of known classes. Formally, for a given data-label pair (x_i, y_i), we expect $\mathbf{f}_i = 1$ and $\mathbf{f}_j = 0$, $\forall j \neq i$. Once such a representation is learned, $\arg\max \mathbf{f}$ returns the most-likely class of an observed sample. Then, $\max \mathbf{f}$ yields the likeliness of the sample belonging to the most likely class. Similar to binary classification, identity I of a test instance can be queried using hard thresholding. In order to learn a representation suitable for the stated objective, we use conventional classification networks as the foundation of our work and propose the following two alternations.

1. Membership loss. Assuming each known class has a unique single label, if the observed image is from a known class, only a single positive activation should appear in $\mathbf{f}$. We observe that when cross-entropy loss is used, this is not the case. To alleviate this, we introduce a new loss called *membership loss* in addition to the cross-entropy loss.

2. Globally negative filters. In a classification setting, a negative filter of a certain class is also a positive filter of another class. In other words, there exist no explicit negative filters. In our formulation, we propose to generate *globally negative filters*

(filters that generate negative evidence for all known classes) to reduce the possibility of a novel sample registering high activation scores.



(a)

(b)

Figure 3-2. (a) Activations of known (Calculator) and unknown samples (Playing Cards) in a VGG16 model.

Limitations of Cross-Entropy Loss

When a classification network is trained, each element f_i of the activation vector $\mathbf{f}$ is first normalized using the softmax function to arrive at a normalized activation vector $\tilde{\mathbf{f}}$ as in, $\tilde{f}_j = e^{f_j} / \sum\limits_{j=1}^{c} e^{f_j}$. When it is assumed that all image classes appearing during inference are known ahead of time, j^{th} element of vector $\tilde{\mathbf{f}}$ is interpreted as the likelihood of the input image x_i belonging to the j^{th} class. Neural network-based classification systems are learned by minimizing the cross-entropy loss which is the negative log likelihood of the correct class $\tilde{\mathbf{f}}$. However, since this is a relative measure, the learned representation deviates from our objective due to the following reasons.

Firstly, even a low activation of the ground truth class could yield a low cross-entropy provided that the activations of all other (non-matching) classes are very low. As a result, lower score values may not get heavily penalized during training. Therefore, a model trained using the cross-entropy loss may end up producing low activation scores for known classes during inference. In closed set classification, this behavior will not cause complications as long as the correct class records the highest score. However, in threshold-based novelty detection, this poses a problem as having low scores for the positive class will result in false negatives. Secondly, the cross-entropy loss does not penalize activations of unrelated classes as long as the correct class produces the highest activation. As a result, inaccurate cross-class relationships are encouraged during training.

In order to illustrate this point, we trained a VGG16 [33] based CNN classification network using the first 128 classes of the Caltech256 dataset. For the considered example, the Calculator class (indexed at 27) is a known class and the Playing Cards class (indexed at 163) is a novel class. Shown in Figure 3-2 are the activations of conv5-3 and fc8 layers of the network for two inputs of the two classes. As can be seen from this figure, when the network observes a calculator object (known object), it correctly associates the highest score in $\mathbf{f}$ to the correct class (class 27). However, there is also a significant miss-association between the calculator class and coin (class 43), keyboard (class 45), dice (class 55) and joystick classes (class 120).

Membership Loss

In our approach, we first independently translate each activation score value f_i into the range $0 - 1$ using the sigmoid(σ) function. We interpret each transformed activation score as the probability of the input image belonging to each individual class. If the ground truth label of a given observation x is y, we aim at learning a function that produces absolute probabilities for the membership of each class as follows

$$\mathbb{P}(y = i) = \sigma(f(x)_i) \;\; \forall i \in \{1, 2, \dots c\}. \tag{3.1}$$

Ideally, the learned transformation will produce $f(x)_i = 1$ for $i = y$ and $f(x)_i = 0$, otherwise. We denote the risk of associating a higher score with a wrong class ($f(x)_i = 1$ for $i \neq y$) as R_{W1} and risk of associating a low score with the correct class ($f(x)_i = 0$ for $i = y$) as R_{C0}. We define the *membership loss* L_M as the risk of classification as

$$L_M(x, y) = R_{C0}(x, y) + \lambda R_{W1}(x, y), \tag{3.2}$$

where λ is a positive scalar. With our formulation, we define $R_{W1}(x, y) = [1 - \mathbb{P}(y = 1)]^2 = [1 - \sigma(f(x)_y)]^2$. Here, the quadratic term is introduced to impose a heavy penalty on very high deviations. Similarly, $R_{C0}(x, y)$ becomes,

$$\begin{aligned} R_{C0}(x, y) &= \frac{1}{c-1} \sum_{i=1, i \neq y}^{c} [\mathbb{P}(i = 1)]^2 \\ &= \frac{1}{c-1} \sum_{i=1, i \neq y}^{c} [\sigma(f(x)_i)]^2. \end{aligned} \tag{3.3}$$

By substitution, we get

$$L_M(x, y) = [1 - \sigma(f(x)_y)]^2 + \lambda \frac{1}{c-1} \sum_{i=1, i \neq y}^{c} [\sigma(f(x)_i)]^2. \tag{3.4}$$

Here, the parameter λ controls relative weight given to each risk source. In our experiments, we set $\lambda = 5$. Taking the partial derivative of the membership loss yields the following back-propagation formula

$$\frac{\partial L_M(x, y)}{\partial f(x)_i} = \begin{cases} -2[1 - \sigma(f(x)_i)] \times \sigma(f(x)_i)' & \text{for } i = y \\ \frac{2\lambda}{c-1} \sigma(f(x)_i) \times \sigma(f(x)_i)' & \text{for } i \neq y, \end{cases} \tag{3.5}$$

where, $\sigma(f(x)_i)' = \sigma(f(x)_i)(1 - \sigma(f(x)_i))$.

The proposed *membership loss* does not operate on the closed-set assumption. It takes individual score values into account in an absolute sense. Therefore, when the membership loss is used, known samples that produce small activations will be penalized regardless of the score values of the other classes. When the membership loss is used together with the cross-entropy loss, the network learns a representation that produces relatively higher activation scores for the correct class. For example, consider the $fc8$ activation map of the proposed method for the Calculator object input shown in Figure 3-2. There, we observe that the correct class (indexed at 27) produces a large positive score whereas all other classes produce negative scores.

Globally Negative Filters

When a conventional classification network is used, novel images are often able to produce very high activation scores there by leading to false positive detections. Such an example is shown in Figure 3-2(bottom) where a Playing Cards instance has produced a very high activation score in the index corresponding to the Calculator class (indexed at 27). Final activation score of a class is generated based on the responses of the positive and negative filters. Once the network is trained, given an input of a particular known class, the input stimulates some *positive filters* and *negative filters* associated with the class. If the model is well trained, the response of the positive filters exceeds the response of the negative filters to produce a high positive activation score.

Given this background, it is interesting to investigate how a novel sample is able to produce a high activation score. Let us revisit activations of Playing Cards image (novel image) shown in Figure 3-2 (bottom). In this example, Playing Cards image has stimulated some positive filters of the Calculator class despite the differences in content. At the same time, by chance, it has not produced sufficient stimulation in negative filters of the Calculator class, thereby producing a large positive activation in $\mathbf{f}$. This can be clearly observed in Figure 3-2 where both the Calculator and the Playing Cards images have activated similar filters in the conv5-3 layer.

To this end, we make the following proposal. We wish to learn a set of filters that are stimulated generally by natural images and produce evidence against all known classes. In other words, these filters are *negative filters* with respect to all known classes - hence we call them *globally negative filters*. If any of such filters are stimulated during inference, it would prove greater evidence that the observed image is novel. However, this proposal will succeed only if the *globally negative filters* are stimulated by arbitrary images outside the known class set.

In order to learn the *globally negative filters*, we propose a joint-learning network structure. In addition to the known object dataset, we use the *out-of-distributional* data samples in training. For the remainder of the paper we refer the *out-of-distributional* dataset as the *reference dataset*. We learn features that can perform classification in both the known dataset and the reference dataset. If the *reference dataset* has $\mathcal{C}$ classes, once trained, the filter bank will contain positive filters of all $c + \mathcal{C}$ classes. Filters associated with the reference dataset will likely act as *negative filters* for all classes in the known dataset, thereby be globally negative. In this framework, the *globally negative filters* are likely to respond to arbitrary natural images provided that the reference dataset is a large-scale diverse dataset.

In Figure 3-2, we show the impact of using the *globally negative filters*. Visualization of top activated filters for the Calculator class are shown at the top in Figure 3-2(b). As can be seen from this figure, these filters are positively co-related with the Calculator class. With the new formulation, we observe that playing cards object activates some extra filters which are not in common with the calculator class (highlighted in red). At the bottom of Figure 3-2(b) we visualize filters with the highest activation for the Playing Cards object. By inspection, these two visualizations look arbitrary and do not have an obvious association with any of the Caltech256 classes. We interpret these filters as instances of the *globally negative filters*. Due to the availability of more negative evidence, the overall activation value of the playing cards object has been drastically reduced.

Training Procedure

We propose a network architecture and a training mechanism to ensure that the network learns the *globally negative filters*. For this purpose, we use an external multi-class labeled dataset which we refer to as the *reference dataset*.

We first select a CNN backbone of choice (this could be a simple network such as Alexnet [32] or a very deep/complex structure such as DenseNet [62]). Two parallel CNN networks of the selected backbone are used for training as shown in Figure 3-3(a). The only difference between the two parallel networks is the final fully-connected layer where the number of outputs is equal to the number of classes present in either dataset.

For the purpose of our discussion, we refer the sub-network up to the penultimate layer of the CNN as the feature extraction sub-network.

Initially, weights of the two feature extraction sub-networks are initialized with identical weights and they are kept identical during training. Final layer of both parallel networks are initialized independently. Weights of these two layers are learned during training without having any dependency between each other. During training, two mini batches from two datasets (reference dataset (R) and known classes (T)) are considered and they are fed into the two branches independently. We calculate the cross-entropy loss (L_{ce}) with respect to the samples of the reference dataset and both the membership loss (L_m) and the cross-entropy loss with respect to the samples of known classes. The cumulative loss of the network then becomes a linear combination of the two losses as follows,

$$CumulativeLoss = L_{ce}(R) + \alpha_1\ L_{ce}(T) + \alpha_2\ L_m(T). \tag{3.6}$$

In our experiments, we keep $\alpha_1, \alpha_2 = 1$. The cumulative loss is back-propagated to learn the weights of the two CNN branches. Reducing membership loss and cross-entropy loss with respect to the known-class dataset increases the potential of performing novelty detection in addition to classification as discussed in the preceding sub-sect. On the other hand, having good performance (low cross-entropy loss) in the reference dataset suggests the existence of filters that are responsive to generic objects provided that the reference dataset is sufficiently diverse. When classes appearing in the reference dataset do not intersect with known classes, these filters serve as the *globally negative filters.*

Testing (Novelty Detection)

During inference, we propose to use the setup shown in Figure 3-3(b) where we only consider the bottom CNN branch of the training network. Given a test image x, we

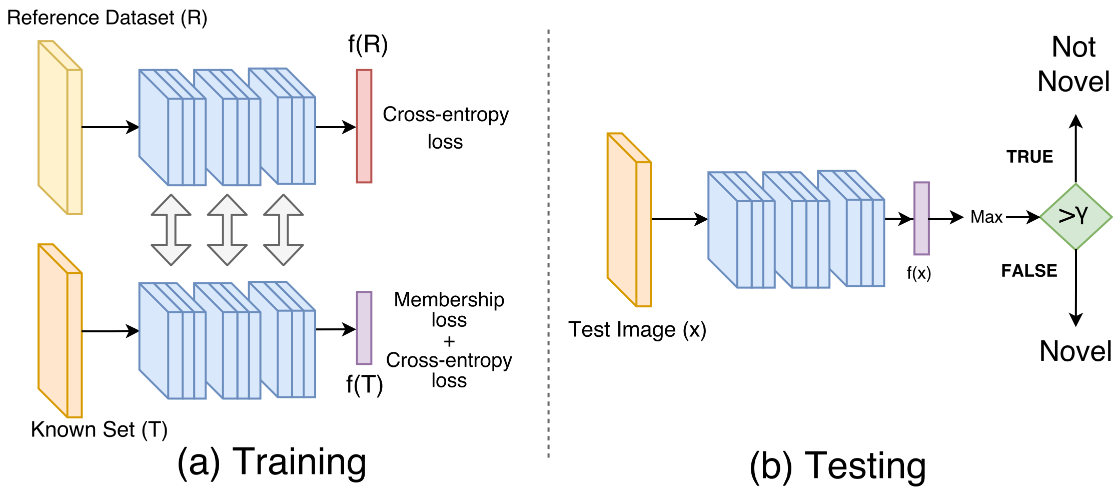


Figure 3-3. Proposed architecture for multiple-class novelty detection with OOD data.

perform a forward pass using the learned CNN network to obtain the final feature $\mathbf{f}(x)$. The largest element of $\mathbf{f}(x)$, $\max \mathbf{f}(x)$ is thresholded using a predetermined threshold γ to arrive at the identity of the test image. If the yielded score is below the threshold γ, we identify the test sample to be novel. In a practical system, threshold γ is chosen considering the percentile of the matched score distribution (for example threshold can be chosen to be 95th percentile if the accepted false negative rate is 5%) . In addition to the novelty detection procedure, the same network structure can be used to perform classification as well. Here, $\arg\max \mathbf{f}(x)$ yields the identity of the predicted class for the test sample x. We note that this step is identical to the classification procedure used in the standard CNN-based classification.

Experimental Setup and Results

In this sect, we present experimental results for the novelty detection task. We first describe the baseline methods used for comparison. Then, we introduce the four datasets used for evaluation. Finally, we discuss the obtained results followed by the analysis of the proposed method.

Baseline Methods

We evaluate the proposed method on four novelty detection databases and we compare its performance with the standard novelty detection schemes. We use the following baseline comparisons based on the AlexNet [32] and the VGG16 [33] features fine-tuned on the given dataset.

1. Finetune [33]**:** $fc8$ feature scores of the trained deep model are thresholded to detect novel samples.

2. One-class SVM [20]**:** A one-class SVM classifier is trained for all known classes. The maximum SVM score is considered during the inference.

3. KNFST [28], [11]**:** Deep features are normalized and histogram intersect kernel method is used to generate inner products between the samples.

4. Local KNFST [51]**:** Deep features with histogram intersect kernel is considered with 600 local regions.

5. OpenMax [49]**:** Activations of penultimate layer of a deep model are used to construct a single channel class-wise mean activation vectors (MAV) and the corresponding Weibull distributions.

6. K-extremes [50]**:** Mean activations of the VGG16 $fc7$ features are considered for each class and top 0.1 activation indexes are binarized to arrive at the Extreme Value Signatures.

7. Finetune$(c+\mathcal{C})$**:** A $(c+\mathcal{C})$ class CNN is trained by treating classes of the reference dataset as the additional class.

In addition, we evaluate the performance based on the pretrained deep features (trained on the ILSVRC12 database) for KNFST and local KNFST methods. Whenever pre-trained features are use they are denoted by the suffix *pre*.

Datasets

We use four publicly available multi-class datasets to evaluate the novelty detection performance of the proposed method.

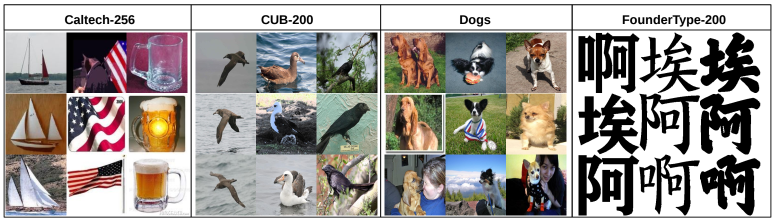


Figure 3-4. Sample images from multiple-class novelty detection evaluation datasets.

Caltech256 Dataset. The Caltech256 dataset is a fully annotated dataset which consists of 30607 images from 256 object classes. Following the protocol presented in [11], we first sorted the class names alphabetically and picked the first 128 classes as the known classes and considered the images from the remaining 128 classes as the novel images.

Caltech-UCSD Birds 200 (CUB 200) Dataset. The CUB-200 dataset includes 6033 images belonging to 200 distinct bird categories. Ground truth labels for each image are provided. In our experiment, we sorted names of the bird categories alphabetically and used the first 100 classes as the known classes. The remaining classes were used to represent novel images.

Stanford Dogs Dataset. This dataset is a subset of the ImageNet dataset and was originally intended for fine-grain classification. There are 20580 images belonging to 120 different dog breeds in this dataset. We considered the first 60 classes as the known classes and treated the remaining classes as the novel classes during performance evaluation.

FounderType-200 Dataset. This dataset is a collection of Chinese character images in different font types. The dataset is organized based on the font-type. In total there are 200 different font-types with 6763 images from each class in this dataset. Following the same convention as before, we picked the first 100 classes to represent the enrolled classes. The remaining 100 classes were used to simulate the novel images.

In all datasets, following the protocol in [11], images of the enrolled classes were randomly split into two even sets to form training and testing datasets of the enrolled classes. Images of the novel classes were used only during testing. When finetuning/extracting features from the caltech256 dataset following [63], we used the pretrained model trained on the Places365 dataset [64]. For all other tasks, we used the pretrained model trained on the ILSVRC12 dataset. Accordingly, the validation sets of Places365 was used as the reference dataset for Caltech256. For all other tasks the validation set of ILSVRC12 was considered.

Results

We evaluated all methods based on the VGG16 and the AlexNet features. We used the training codes provided by the authors when evaluating the KNFST [28] and the local KNFST [51] methods. Performance of each method is evaluated using the area under the receiver operating characteristics (AUC) curve. Obtained AUC values for each method are tabulated in Table 3-I for all datasets[1].

When baseline methods are considered, a variance in performance can be observed across datasets. In general, K-extremes has reported below-par performances compared to the other methods. When the number of enrolled classes are very high, the mean activation signature of a class looses its uniqueness. This is why K-extremes method fails when very large number of classes are enrolled as suggested in [50]. In the Caltech-256 and CUB-200 datasets, thresholding deep activations and OpenMax has yielded better results among the baseline methods. In Caltech256, this has improved marginally when the reference dataset (ILSVRC12) is incorporated. This method has performed reasonably well in the FounderType-200 dataset but it's performance in

---

[1] Source code of the proposed method is made available at https://github.com/PramuPerera/TransferLearningNovelty

Table 3-I. Novelty detection results (AUC of the ROC curve) on the evaluation datasets. The best performing method for each dataset is shown in bold. Second best method is shown in italics.

Method	Caltech-256		CUB-200		Dogs		FounderType	
	VGG16	AlexNet	VGG16	AlexNet	VGG16	AlexNet	VGG16	AlexNet
Finetune[33], [32]	0.827	0.785	0.931	0.909	0.766	0.702	0.841	0.650
One-class SVM[20]	0.576	0.561	0.554	0.532	0.542	0.520	0.627	0.612
KNFST pre[28]	0.727	0.672	0.842	0.710	0.649	0.619	0.590	0.655
KNFST[28], [11]	0.743	0.688	0.891	0.748	0.633	0.602	*0.870*	0.678
Local KNFST pre[51]	0.657	0.600	0.780	0.717	0.652	0.589	0.549	0.523
Local KNFST[51]	0.712	0.628	0.820	0.690	0.626	0.600	0.673	0.633
K-extremes[50]	0.546	0.521	0.520	0.514	0.610	0.592	0.557	0.512
OpenMax[49]	0.831	0.787	*0.935*	*0.915*	0.776	*0.711*	0.852	0.667
Finetune($c + \mathcal{C}$)	*0.848*	*0.788*	0.921	0.899	*0.780*	0.692	0.754	*0.723*
Deep Novelty (ours)	**0.869**	**0.807**	**0.958**	**0.947**	**0.825**	**0.748**	**0.893**	**0.741**

the Standford Dogs dataset is not convincing. In general, KNFST has out-performed local KNFST except for in the Standford Dogs dataset. KNFST (and local KNFST) operating on the finetuned deep features have performed better in general compared to the pre-trained deep features. This trend has changed only in the Standford Dogs dataset. Here we note that none of the baseline methods have yielded consistent performance across datasets.

In comparison, the proposed method is able to produce the best performance across all datasets. When AlexNet is used as the back-bone network, there is an improvement of about 3.0% over the baselines in the CUB-200 and Standford Dogs datasets. In the other two datasets this margin is 2.0%. In the Caltech256, CUB-200 and FounderType-200 datasets, the improvements in AUC are in excess of 2.0% for the VGG16 model. In the Standford Dogs dataset, the proposed method is able to introduce a significant advancement of more than 7.0% in AUC compared with the baseline methods. In general, we note that in datasets where the baseline performance is already very good, as in the CUB-200 and FounderType 200 datasets, the improvement of the proposed method is relatively small. On the other hand, when the baseline performance is poor, the proposed method is able to generate a significant improvement in the performance.

Ablation Study

In this subsect, we investigate the impact of each individual component of the proposed framework. For the purpose of the ablation study, we use the validation dataset of the ILSVRC12 dataset as the reference dataset. It should be noted that figures reported in this subsect are different from Table 3-I due to this reason. Starting from the traditional CNN architecture, we added one component of the proposed framework at a time and evaluated the novelty detection performance on the Caltech-256 dataset as a case study. Testing protocol presented in the preceding subsect was followed in all cases. Considered cases are as follows.

a) Single CNN with the cross-entropy loss (AUC 0.854). This is the CNN baseline where a CNN is trained using the enrolled classes conventionally.

b) Single CNN with the cross-entropy loss+membership loss (AUC 0.865). The network architecture is the same as in case (a). In addition to the cross-entropy loss, the membership loss is calculated with respect to the enrolled dataset.

c) Two Parallel CNNs with cross-entropy loss (AUC 0.864). The network structure proposed in Figure 3-3(a) is used. In contrast, only the cross-entropy loss is used in the bottom sub-network.

d) Proposed method (AUC 0.906). Proposed structure Figure 3-3(a) is used for training.

In the proposed method, we introduced membership loss and a parallel network structure as contributions. From the case study conducted, it appears that the novelty detection performance improves compared to the baseline even when one of the contributions are used. Moreover, we observe that the two contributions compliment each other and generate even better results when combined together.

Impact of the Reference Dataset

In the proposed method, we assumed the availability of a reference dataset with large number of classes. In this subsect, we investigate the impact of the reference dataset by varying the reference dataset of choice. In particular, we use the ILSVRC12, Caltech-256 and Standford Dogs datasets as the reference datasets to perform novelty detection using the proposed method in the CUB-200 dataset. Results obtained are tabulated in Table 3-II. Here we have included the performance of the best baseline method for the CUB-200 dataset (Finetune) from Table 3-I as a baseline.

Compared to ILSVRC12, when Caltech-256 is used as the reference dataset, AUC drops by 0.005%. This further drops by 0.008% when the Standford Dogs dataset is used. The ILSVRC12 dataset contains 1000 image classes and has significant variance in images within each class. Caltech-256 is a similar multi-class dataset but with fewer classes. Both of these datasets contain natural images. However since ILSVRC12 has more classes and more intra-class variance, we expect it to generate *globally negative filters* better. Therefore, the performance drop of Caltech-256 compared to ILSVRC12 is expected. On the other hand, the Standford Dogs dataset only contains images of dogs. Therefore, filters learned using this dataset may not be generic to get stimulated by the arbitrary inputs. Therefore, the drop in the performance is justified. In conclusion, we note that the proposed method is able to out-perform baseline novelty detection methods even when the reference dataset is varied. However, better results are obtained when a larger dataset with high degree of intra-class variation is used as the reference dataset.

Table 3-II. Impact of the reference dataset used. Results of the case study conducted on the CUB-200 dataset by varying the reference dataset.

	Baseline	ILSVRC12	Caltech-256	Dogs
Novelty Detection AUC	0.931	**0.958**	0.953	0.945

Impact on Classification Accuracy

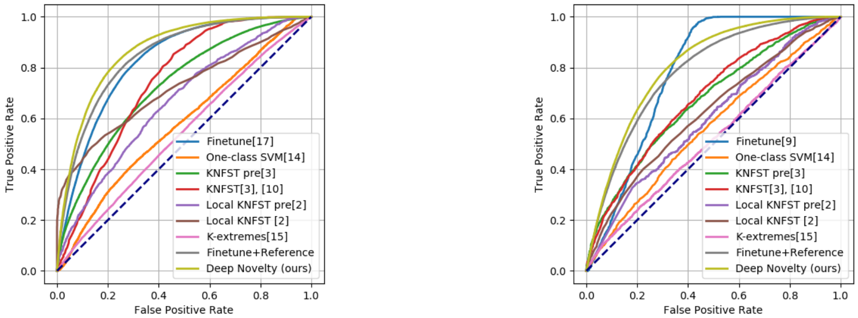
When a test image is present, the proposed method produces a set of class activation scores. It is still possible to perform classification using the same system by associating the test image with the class containing the highest activation. In what follows, we consider test samples of the known classes and perform closed-set classification in the same experimental setup described in sect 7. In other words, we do not consider novel samples for the purpose of this study. Obtained classification accuracies for the four datasets are tabulated in Table 3-III. Although the proposed method is designed for the purpose of novelty detection, we note that the proposed changes have contributed towards increasing the classification accuracy of the system as well. This is because the *membership loss* explicitly enforces correct class to have a high score and all other classes to have scores closer to zero.

Table 3-III. Classification accuracy obtained for conventional fine-tuning and the proposed method for the four evaluation datasets.

	Caltech-256	CUB-200	Dogs	FounderType
VGG16	0.908	0.988	0.730	0.945
Proposed Method	**0.939**	**0.990**	**0.801**	**0.950**

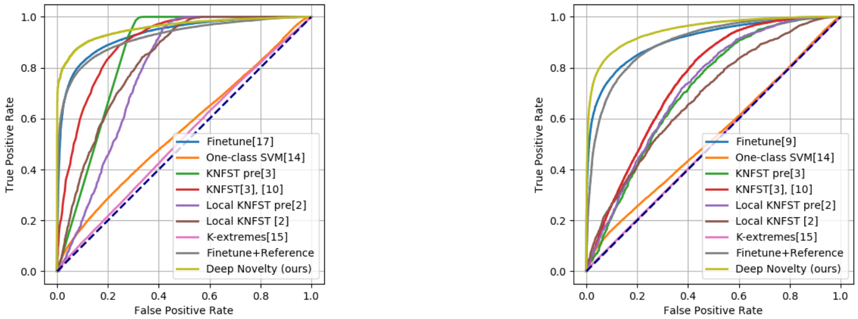
Network Hyper-parameters

For all experiments (networks A-C) a batch size of 32 was used with a weight decay of 0.0005. Stochastic Gradient Descent was used as the solver. For all datasets except for the Standford Dogs dataset, base learning rate was set to 0.001. For the Standford Dogs dataset base learning rate was set to 0.0001 to prevent exploding gradients. In all cases, learning rate decay policy with a factor of 0.1 for each 10000 iterations was used. For the VGG16-based networks, all layers up to Conv5-3 were fixed during training. For the AlexNet-based networks, all convolutional layers were fixed during training.



VGG16 *AlexNet*

Figure 3-5. ROC curves obtained for novelty detection in the Caltech256 dataset.



VGG16 *AlexNet*

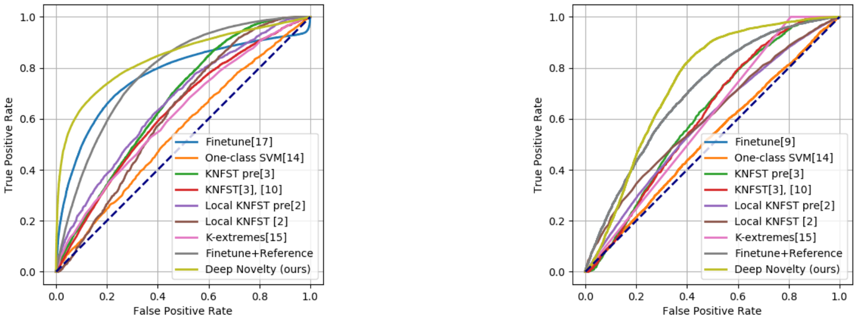
Figure 3-6. ROC curves obtained for novelty detection in the CUB200 dataset.

ROC Curves

In this subsection, we present the ROC curves obtained in each experiment. The ROC curves obtained for the Caltech256, CUB200, FounderType200 and Standford Dogs datasets are shown in Figures 1-4, respectively. It can be seen from these figures, the proposed method has obtained the best ROC curves in all cases.

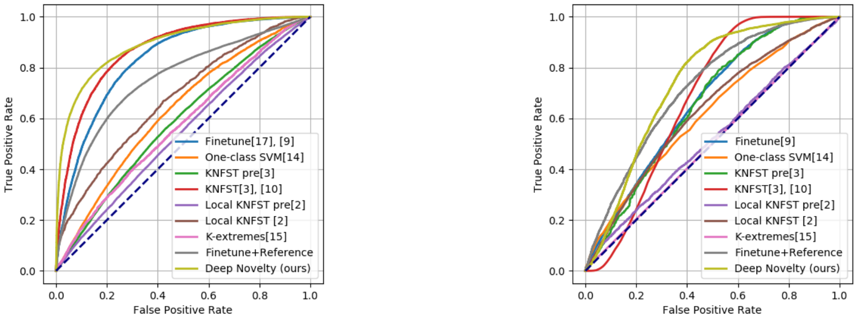
Summary

A deep learning-based solution that takes advantage of out of distribution data was presented targetting multiple-class novelty detection applications. We build up on the conventional classification networks and introduce two novel contributions; namely, *membership loss* and a training procedure that produces *globally negative filters*. In



VGG16 *AlexNet*

Figure 3-7. ROC curves obtained for novelty detection in the FounderType200 dataset.



VGG16 *AlexNet*

Figure 3-8. ROC curves obtained for novelty detection in the Stanford Dogs dataset.

the proposed method, novelty is quarried simply by thresholding the highest activation of the output vector. We demonstrate the effectiveness of the proposed method on four publicly available multi-class image datasets and obtain state-of-the-art results.

Chapter 4

One-class Novelty Detection with OOD Data

Contemporary one-class classification schemes trained solely on the given concept have failed to produce promising results in real-world datasets ([23],[12] has achieved an Area Under the Curve in the range of 60%-65% for CIFAR10 dataset[65]). However, we note that computer vision is a field rich with labeled datasets of different domains. In this chapter, we investigate how data from a different domain can be used to solve the one-class classification problem.

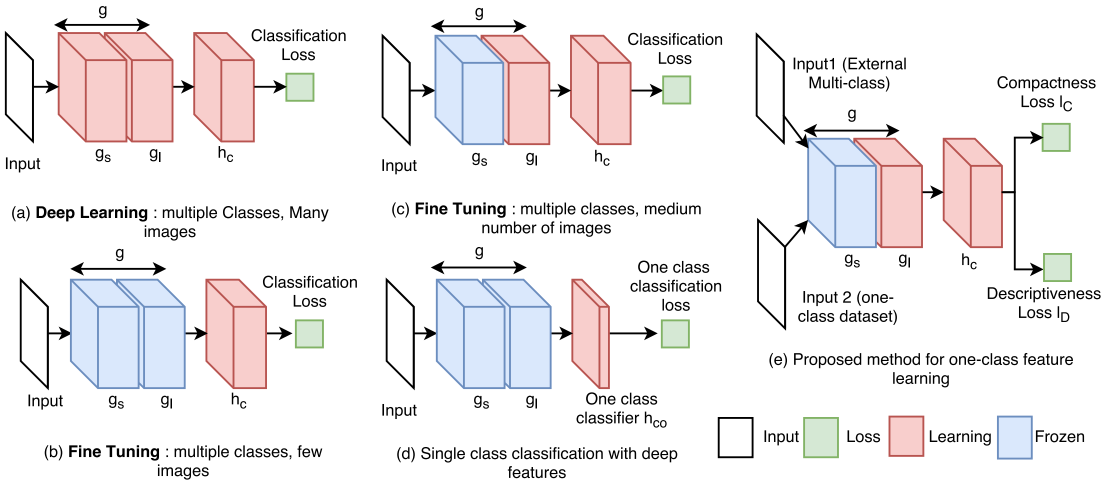
In order to solve this problem, we seek motivation from generic object classification frameworks. Many previous works in object classification have focused on improving either the feature or the classifier (or in some cases both) in an attempt to improve the classification performance. In particular, various deep learning-based feature extraction and classification methods have been proposed in the literature and have gained a lot of traction in recent years [32], [33]. In general, deep learning-based classification schemes have two subnetworks, a feature extraction network (g) followed by a classification sub network (h), that are learned jointly during training. For example, in the popular AlexNet architecture [32], the collection of convolution layers may be regarded as (g) where as fully connected layers may collectively be regarded as (h). Depending on the output of the classification sub network (h), one or more losses are evaluated to facilitate training. Deep learning requires the availability of multiple classes for training and extremely large number of training samples (in the order of thousands or millions). However, in learning tasks where either of these conditions are not met, the following alternative strategies are used:

(a) Multiple classes, many training samples: This is the case where both requirements are satisfied. Both feature extraction and classification networks, g and h are trained end-to-end (Figure 4-1(a)). The network parameters are initialized using random weights. Resultant model is used as the pre-trained model for fine tuning [32], [59].

(b) Multiple classes, low to medium number of training samples: The feature extraction network from a pre-trained model is used. Only a new classification network is trained in the case of low training samples (Figure 4-1(b)). When medium number of training samples are available, feature extraction network (g) is divided into two sub-networks - shared feature network (g_s) and learned feature network (g_l), where $g = g_s \circ g_l$. Here, g_s is taken from a pre-trained model. g_l and the classifier are learned from the data in an end-to-end fashion (Figure 4-1(c)). This strategy is often referred to as fine-tuning [66].

(c) Single class or no training data: A pre-trained model is used to extract features. The pre-trained model used here could be a model trained from scratch (as in (a)) or a model resulting from fine-tuning (as in (b)) [35], [8] where training/fine-tuning is performed based on an external dataset. When training data from a class is available, a one-class classifier is trained on the extracted features (Figure 4-1(d)). It is possible to consider features from the output of the classification sub-network h or features from an intermediate layer of the classification sub-network. To be generic, we denote the collection of layers used for feature extraction as h_c for the remainder of this chapter.

In this work, we focus on the task presented in case (c) where training data from


g
Classification Loss
Input
g_s
g_l
h_c
One class classification loss
One class classifier h_{co}
Input1 (External Multi-class)
Input 2 (one-class dataset)
Compactness Loss l_C
Descriptiveness Loss l_D
Input
Loss
Learning
Frozen

(a) **Deep Learning** : multiple Classes, Many images

(b) **Fine Tuning** : multiple classes, few images

(c) **Fine Tuning** : multiple classes, medium number of images

(d) Single class classification with deep features

(e) Proposed method for one-class feature learning

Figure 4-1. Different deep learning strategies used for classification.

a single class is available. Strategy used in case (c) above uses deep-features extracted from a pre-trained model, where training is carried out on a different dataset, to perform one-class classification. However, there is no guarantee that features extracted in this fashion will be as effective in the new one-class classification task. In this work, we present a feature fine tuning framework which produces deep features that are specialized to the task at hand. Once the features are extracted, they can be used to perform classification using the strategy discussed in (c).

In our formulation (shown in Figure 4-1 (e)), starting from a pre-trained deep model, we freeze initial features (g_s) and learn (g_l) and (h_c). Based on the output of the classification sub-network (h_c), two losses *compactness loss* and *descriptiveness loss* are evaluated. These two losses, introduced in the subsequent sections, are used to assess the quality of the learned deep feature. We use the provided one-class dataset to calculate the *compactness loss*. An external multi-class reference dataset is used to evaluate the *descriptiveness loss*. As shown in Figure 4-2, weights of g_l and h_c are learned in the proposed method through back-propagation from the composite loss. Once training is converged, system shown in setup in Figure 4-1(d) is used to perform classification where the resulting model is used as the pre-trained model.

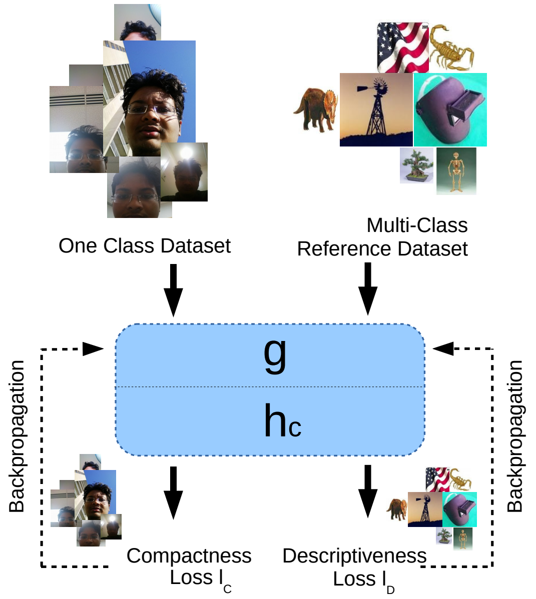


Figure 4-2. Overview of the proposed method.

Objective Function

In this section, we formulate the objective of one-class feature learning as an optimization problem. In the classical multiple-class classification, features are learned with the objective of maximizing inter-class distances between classes and minimizing intra-class variances within classes [67]. However, in the absence of multiple classes such a discriminative approach is not possible.

In this light, we outline two important characteristics of features intended for one-class classification.

Compactness $\mathcal{C}$. A desired quality of a feature is to have a similar feature representation for different images of the same class. Hence, a collection of features extracted from a set of images of a given class will be compactly placed in the feature space. This quality is desired even in features used for multi-class classification. In such cases, compactness is quantified using the intra-class distance [67]; a compact representation would have a lower intra-class distance.

Descriptiveness $\mathcal{D}$. The given feature should produce distinct representations for images of different classes. Ideally, each class will have a distinct feature representation from each other. Descriptiveness in the feature is also a desired quality in multi-class classification. There, a descriptive feature would have large inter-class distance [67].

It should be noted that for a useful (discriminative) feature, both of these characteristics should be satisfied collectively. Unless mutually satisfied, neither of the above criteria would result in a useful feature. With this requirement in hand, we aim to find a feature representation g that maximizes both compactness and descriptiveness. Formally, this can be stated as an optimization problem as follows,

$$\hat{g} = \max_{g} \ \mathcal{D}(g(t)) + \lambda \mathcal{C}(g(t)), \tag{4.1}$$

where t is the training data corresponding to the given class and λ is a positive constant. Given this formulation, we identify three potential strategies that may be employed when deep learning is used for one-class problems. However, none of these strategies collectively satisfy both descriptiveness and compactness.

(a) Extracting deep features. Deep features are first extracted from a pre-trained deep model for given training images. Classification is done using a one-class classification method such as one-class SVM, SVDD or k-nearest neighbor using extracted features. This approach does not directly address the two characteristics of one-class features. However, if the pre-trained model used to extract deep features was trained on a dataset with large number of classes, then resulting deep features are likely to be descriptive. Nevertheless, there is no guarantee that the used deep feature will possess the compactness property.

(b) Fine-tune a two class classifier using an external dataset. Pre-trained deep networks are trained based on some legacy dataset. For example, models used for the ImageNet challenge are trained based on the ImageNet dataset [63]. It is possible to fine tune the model by representing the alien classes using the legacy dataset. This strategy will only work when there is a high correlation between alien classes

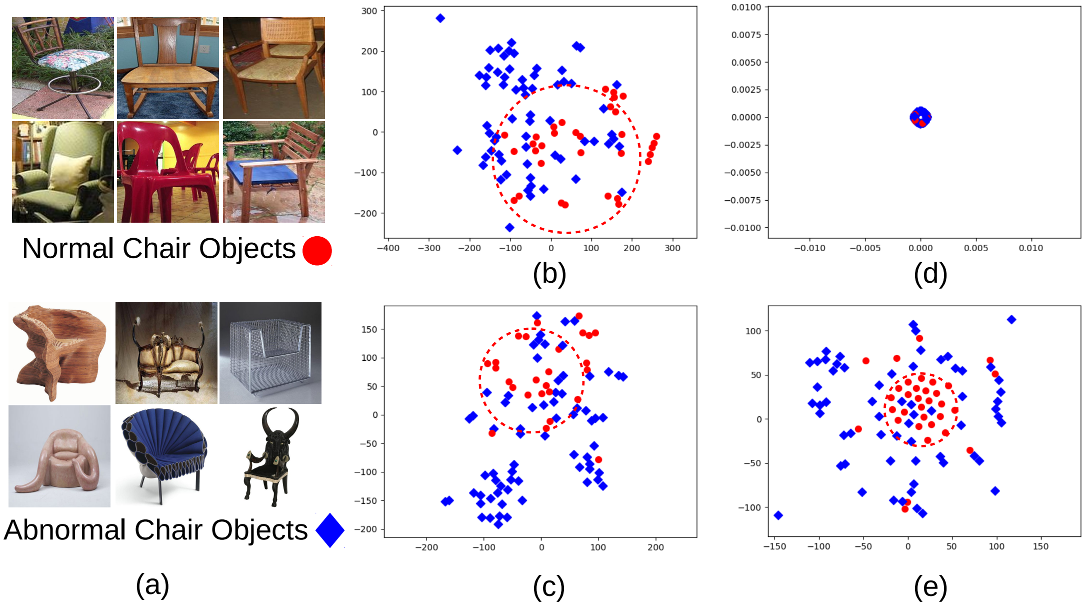


Figure 4-3. Possible strategies for one-class classification in abnormal image detection. (a)Image samples. (b) AlexNet features. (c) Binary CNN (d) Fine-tuning (e) Proposed method.

and the legacy dataset. Otherwise, the learned feature will not have the capacity to describe the difference between a given class and the alien class thereby violating the descriptiveness property.

(c) Fine-tune using a single class data. Fine-tuning may be attempted by using data only from the given single class. For this purpose, minimization of the traditional cross-entropy loss or any other appropriate distance could be used. However, in such a scenario, the network may end up learning a trivial solution due to the absence of a penalty for miss-classification. In this case, the learned representation will be compact but will not be descriptive.

Let us investigate the appropriateness of these three strategies by conducting a case study on the abnormal image detection problem where the considered class is the *normal chair* class. In abnormal image detection, initially a set of *normal chair* images are provided for training as shown in Figure 4-3(a). The goal is to learn a representation such that, it is possible to distinguish a *normal chair* from an *abnormal* *chair.*

The trivial approach to this problem is to extract deep features from an existing CNN architecture (solution (a)). Let us assume that the AlexNet architecture [32] is used for this purpose and *fc7* features are extracted from each sample. Since deep features are sufficiently descriptive, it is reasonable to expect samples of the same class to be clustered together in the extracted feature space. Illustrated in Figure 4-3(b) is a 2D visualization of the extracted 4096 dimensional features using t-SNE [68]. As can be seen from Figure4-3(b), the AlexNet features are not able to enforce sufficient separation between normal and abnormal chair classes.

Another possibility is to train a two class classifier using the AlexNet architecture by providing normal chair object images and the images from the ImageNet dataset as the two classes (solution (b)). However, features learned in this fashion produce similar representations for both normal and abnormal images, as shown in Figure4-3(c). Even though there exist subtle differences between normal and abnormal chair images, they have more similarities compared to the other ImageNet objects/images. This is the main reason why both normal and abnormal images end up learning similar feature representations.

A naive, and ineffective, approach would be to fine-tune the pre-trained AlexNet network using only the *normal chair* class (solution (c)). Doing so, in theory, should result in a representation where all *normal chairs* are compactly localized in the feature space. However, since all class labels would be identical in such a scenario, the fine-tuning process would end up learning a futile representation as shown in Figure4-3(d). The reason why this approach ends up yielding a trivial solution is due to the absence of a regularizing term in the loss function that takes into account the discriminative ability of the network. For example, since all class labels are identical, a zero loss can be obtained by making all weights equal to zero. It is true that this is a valid solution in the closed world where only *normal chair* objects exist. But such a network has zero discriminative ability when *abnormal chair* objects appear.

None of the three strategies discussed above are able to produce features that are both compact and descriptive. We note that out of the three strategies, the first produces the most reasonable representation for one-class classification. However, this representation was learned without making an attempt to increase compactness of the learned feature. Therefore, we argue that if compactness is taken into account along with descriptiveness, it is possible to learn a more effective representation.

Proposed Loss Functions

In this work, we propose to quantify compactness and descriptiveness in terms of measurable loss functions. Variance of a distribution has been widely used in the literature as a measure of the distribution spread [69]. Since spread of the distribution is inversely proportional to the compactness of the distribution, it is a natural choice to use variance of the distribution to quantify compactness. In our work, we approximate variance of the feature distribution by the variance of each feature batch. We term this quantity as the *compactness loss* (l_C).

On the other hand, descriptiveness of the learned feature cannot be assessed using a single class training data. However, if there exists a reference dataset with multiple classes, even with random object classes unrelated to the problem at hand, it can be used to assess the descriptiveness of the engineered feature. In other words, if the learned feature is able to perform classification with high accuracy on a different task, the descriptiveness of the learned feature is high. Based on this rationale, we use the learned feature to perform classification on an external multi-class dataset, and consider classification loss there as an indicator of the descriptiveness of the learned feature. We call the cross-entropy loss calculated in this fashion as the *descriptiveness loss* (l_D). Here, we note that *descriptiveness loss* is low for a descriptive representation.

With this formulation, the original optimization objective in equation (4.1) can be re-formulated as,

$$\hat{g} = \min_{g} \ l_D(r) + \lambda l_C(t), \tag{4.2}$$

where l_C and l_D are *compactness loss* and *descriptiveness loss*, respectively and r is the training data corresponding to the reference dataset. The tSNE visualization of the features learned in this manner for normal and abnormal images are shown in Figure 4-3(e). Qualitatively, features learned by the proposed method facilitate better distinction between normal and abnormal images as compared with the cases is shown in Figure 4-1(b)-(d).

Terminology

Based on the intuition given in the previous section, the architecture shown in Figure 4-4 (a) is proposed for one-class classification training and the setup shown in Figure 4-4 (b) for testing. They consist of following elements:

Reference Network (R): This is a pre-trained network architecture considered for the application. Typically it contains a repetition of convolutional, normalization, and pooling layers (possibly with skip connections) and is terminated by an optional set of fully connected layers. For example, this could be the AlexNet network [32] pre-trained using the ImageNet [63] dataset. Reference network can be seen as the composition of a feature extraction sub-network g and a classification sub-network h_c. For example, in AlexNet, *conv1-fc7* layers can be associated with g and fc8 layer with h_c. *Descriptive loss* (l_D) is calculated based on the output of h_c.

Reference Dataset (r): This is the dataset (or a subset of it) used to train the network R. Based on the example given, reference dataset is the ImageNet dataset [63] (or just a subset of the ImageNet dataset).

Secondary Network (S): This is a second CNN where the network architecture is structurally identical to the reference network. Note that g and h_c are shared in both of these networks. *Compactness loss* (l_C) is evaluated based on the output of h_c. For

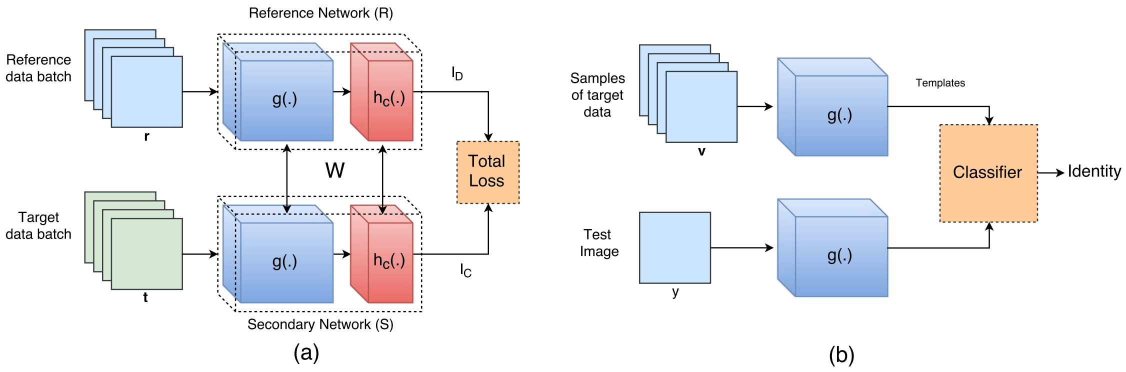


Figure 4-4. (a) Training, and (b) testing frameworks of the proposed DOC method.

the considered example, S would have the same structure as R (AlexNet) up to *fc8*.

Target Dataset (t): This dataset contains samples of the class for which one-class learning is used for. For example, for an abnormal image detection application, this dataset will contain normal images (i.e. data samples of the single class considered).

Model (W): This corresponds to the collection of weights and biases in the network, g and h_c. Initially, it is initialized by some pre-trained parameters W_0. [1]

Compactness loss (l_C) : All the data used during the training phase will belong to the same class. Therefore they all share the same class label. *Compactness loss* evaluates the average similarity between the constituent samples of a given batch. For a large enough batch, this quantity can be expected to represent average intra-class variance of a given class. It is desirable to select a smooth differentiable function as l_C to facilitate back propagation. In our work, we define compactness loss based on the Euclidean distance.

Descriptiveness loss (l_D) : Descriptiveness loss evaluates the capacity of the learned feature to describe different concepts. We propose to quantify discriminative loss by the evaluation of cross-entropy with respect to the reference dataset (R).

For this discussion, we considered the AlexNet CNN architecture as the reference

---

[1] For the case of AlexNet, pre-trained model can be found at www.berkleyvison.com.

network. However, the discussed principles and procedures would also apply to any other CNN of choice. In what follows, we present the implementation details of the proposed method.

Architecture

The proposed training architecture is shown in Figure 4-4 (a) [2]. The architecture consists of two CNNs, the reference network (R) and the secondary network (S) as introduced in the previous sub-section. Here, weights of the reference network and secondary network are tied across each corresponding counterparts. For example, weights between conv$i$ (where, $i = 1, 2.., 5$) layer of the two networks are tied together forcing them to be identical. All components, except *Compactness loss*, are standard CNN components. We denote the common feature extraction sub-architecture by $g(.)$ and the common classification by sub-architecture by $h_c(.)$. Please refer to Appendix for more details on the architectures of the proposed method based on the AlexNet and VGG16 networks.

Compactness loss

Compactness loss computes the mean squared intra-batch distance within a given batch. In principle, it is possible to select any distance measure for this purpose. In our work, we design compactness loss based on the Euclidean distance measure. Define $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\} \in R^{n \times k}$ to be the input to the loss function, where the batch size of the input is n.

Forward Pass: For each i^{th} sample $\mathbf{x_i} \in \mathbb{R}^k$, where $1 \leq i \leq n$, the distance between the given sample and the rest of the samples $\mathbf{z_i}$ can be defined as,

$$\mathbf{z_i} = \mathbf{x_i} - \mathbf{m_i}, \tag{4.3}$$

---

[2] Source code of the proposed method is made available online at https://github.com/PramuPerera/DeepOneClass

where, $\mathbf{m_i} = \frac{1}{n-1}\sum_{j \neq i} \mathbf{x_j}$ is the mean of rest of the samples. Then, compactness loss l_C is defined as the average Euclidean distance as in,

$$l_C = \frac{1}{nk}\sum_{i=1}^{n} \mathbf{z_i}^T \mathbf{z_i}. \tag{4.4}$$

Backpropagation:. In order to perform back-propagation using this loss, it is necessary to assess the contribution each element of the input has on the final loss. Let $\mathbf{x_i} = \{x_{i1}, x_{i2}, \dots, x_{ik}\}$. Similarly, let $\mathbf{m_i} = \{m_{i1}, m_{i2}, \dots, m_{ik}\}$. Then, the gradient of l_b with respect to the input x_{ij} is given as,

$$\frac{\partial l_C}{\partial x_{ij}} = \frac{2}{(n-1)nk}\Bigg[n \times (x_{ij} - m_{ij}) - \sum_{k=1}^{n}(x_{ik} - m_{ik})\Bigg]. \tag{4.5}$$

Detailed derivation of the back-propagation formula can be found in the Appendix. The loss l_C calculated in this form is equal to the sample feature variance of the batch multiplied by a constant (see Appendix). Therefore, it is an inverse measure of the compactness of the feature distribution.

Training

During the training phase, initially, both the reference network (R) and the secondary network (S) are initialized with the pre-trained model weights $\mathbf{W_0}$. Recall that except for the types of loss functions associated with the output, both these networks are identical in structure. Therefore, it is possible to initialize both networks with identical weights. During training, all layers of the network except for the last four layers are frozen as commonly done in network fine-tuning. In addition, the learning rate of the training process is set at a relatively lower value (5×10^{-5} is used in experiments). During training, two image batches, each from the reference dataset and the target dataset are simultaneously fed into the input layers of the reference network and secondary network, respectively. At the end of the forward pass, the reference network generates a *descriptiveness loss* (l_D), which is same as the cross-entropy loss by definition, and the secondary network generates *compactness loss* (l_C). The composite loss (l) of the network is defined as,

$$l(r,t) = l_D(r|W) + \lambda l_C(t|W), \tag{4.6}$$

where λ is a constant. It should be noted that, minimizing this loss function leads to the minimization of the optimization objective in (4.2).

In our experiments, λ is set equal to 0.1 Based on the composite loss, the network is back-propagated and the network parameters are learned using gradient descent or a suitable variant. Training is carried out until composite loss $l(r,t)$ converges. A sample variation of training loss is shown in Figure 4-5. In general, it was observed that composite loss converged in around two epochs (here, epochs are defined based on the size of the target dataset).

Intuitively, the two terms of the loss function l_D and l_C measure two aspects of features that are useful for one-class learning. Cross entropy loss values obtained in calculating *descriptiveness loss* l_D measures the ability of the learned feature to describe different concepts with respect to the reference dataset. Having reasonably good performance in the reference dataset implies that the learned features are discriminative in that domain. Therefore, they are likely to be descriptive in general. On the other hand, *compactness loss* (l_C) measures how compact the class under consideration is in the learned feature space. The weight λ governs the mutual importance placed on each requirement.

If λ is made large, it implies that the descriptiveness of the feature is not as important as the compactness. However, this is not a recommended policy for one-class learning as doing so would result in trivial features where the overlap between the given class and an alien class is significantly high. As an extreme case, if $\lambda = 0$ (this is equivalent to removing the reference network and carrying out training solely on the secondary network (Figure 4-1 (d)), the network will learn a trivial solution. In our experiments, we found that in this case the weights of the learned filters become zero thereby making output of any input equal to zero.

Therefore, for practical one-class learning problems, both reference and secondary networks should be present and more prominence should be given to the loss of the reference network.

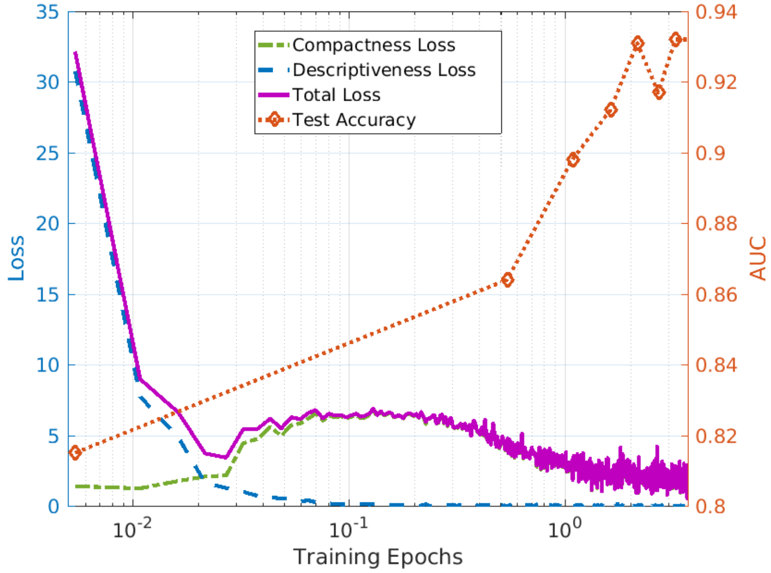


Figure 4-5. Variation of loss functions during training in the proposed method.

Testing

The proposed testing procedure involves two phases - template generation and matching. For both phases, secondary network with weights learned during training is used as shown in Figure 4-4 (b). During both phases, the excitation map of the feature extraction sub-network is used as the feature. For example, layer 7, *fc7* can be considered from a AlexNet-based network. First, during the template generation phase a small set of samples $v = \{v_1, v_2, \dots, v_n\}$ are drawn from the target (i.e. training) dataset where $v \in t$. Then, based on the drawn samples a set of features $g(v_1), g(v_2), \dots, g(v_n)$ are extracted. These extracted features are stored as templates and will be used in the matching phase.

Based on stored template, a suitable one-class classifier, such as one-class SVM [20], SVDD [21] or k-nearest neighbor, can be trained on the templates. In this work, we choose the simple k-nearest neighbor classifier described below. When a test image y is present, the corresponding deep feature $g(y)$ is extracted using the described method. Here, given a set of templates, a matched score S_y is assigned to y as

$$S_y = f(g(y)|g(t_1), g(t_2), \ldots, g(t_n)), \tag{4.7}$$

where $f(.)$ is a matching function. This matching function can be a simple rule such as the cosine distance or it could be a more complicated function such as Mahalanobis distance. In our experiments, we used Euclidean distance as the matching function. After evaluating the matched score, y can be classified as follows,

$$class(y) = \begin{cases} 1, & \text{if } S_y \leq \delta \\ 0, & \text{if } S_y > \delta, \end{cases} \tag{4.8}$$

where 1 is the class identity of the class under consideration and 0 is the identity of other classes and δ is a threshold.

Memory Efficient Implementation

Due to shared weights between the reference network and the secondary network, the amount of memory required to store the network is nearly twice as the number of parameters. It is not possible to take advantage of this fact with deep frameworks with static network architectures (such as caffe [70]). However, when frameworks that support dynamic network structures are used (e.g. PyTorch), implementation can be altered to reduce memory consumption of the network.

In the alternative implementation, only a single core network with functions g and h_c is used. Two loss functions l_C and l_D branch out from the core network. However in this setup, *descriptiveness loss (l_D)* is scaled by a factor of $1 - \lambda$. In this formulation, first λ is made equal to 0 and a data batch from the reference dataset is fed into the network. Corresponding loss is calculated and resulting gradients are calculated using back-propagation Then, λ is made equal to 1 and a data batch from the target dataset is fed into the network. Gradients are recorded same as before after back-propagation. Finally, the average gradient is calculated using two recorded gradient values, and network parameters are updated accordingly. In principle, despite of having a lower memory requirement, learning behavior in the alternative implementation would be identical to the original formulation.

Experimental Results

In order to asses the effectiveness of the proposed method, we consider three one-class classification tasks: abnormal image detection, single class image novelty detection and active authentication. We evaluate the performance of the proposed method in all three cases against state of the art methods using publicly available datasets. Further, we provide two additional CNN-based baseline comparisons.

Experimental Setup

Unless otherwise specified, we used 50% of the data for training and the remaining data samples for testing. In all cases, 40 samples were taken at random from the training set to generate templates. In datasets with multiple classes, testing was done by treating one class at a time as the positive class. Objects of all the other classes were considered to be alien. During testing, alien object set was randomly sampled to arrive at equal number of samples as the positive class. As for the reference dataset, we used the validation set of the ImageNet dataset for all tests. When there was an object class overlap between the target dataset and the reference dataset, the corresponding overlapping classes were removed from the reference dataset. For example, when novelty detection was performed based on the Caltech 256, classes appearing in both

Caltech 256 and ImageNet were removed from the ImageNet dataset prior to training.

The Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) Curve are used to measure the performance of different methods. The reported performance figures in this paper are the average AUC figures obtained by considering multiple classes available in the dataset. In all of our experiments, Euclidean distance was used to evaluate the similarity between a test image and the stored templates. In all experiments, the performance of the proposed method was evaluated based on both the AlexNet [32] and the VGG16 [33] architectures. In all experimental tasks, the following experiments were conducted.

AlexNet Features and VGG16 Features (Baseline). One-class classification is performed using k-nearest neighbor, One-class SVM[20], Isolation Forest[13] and Gaussian Mixture Model[13] classifiers on $fc7$ AlexNet features and the $fc7$ VGG16 features, respectively.

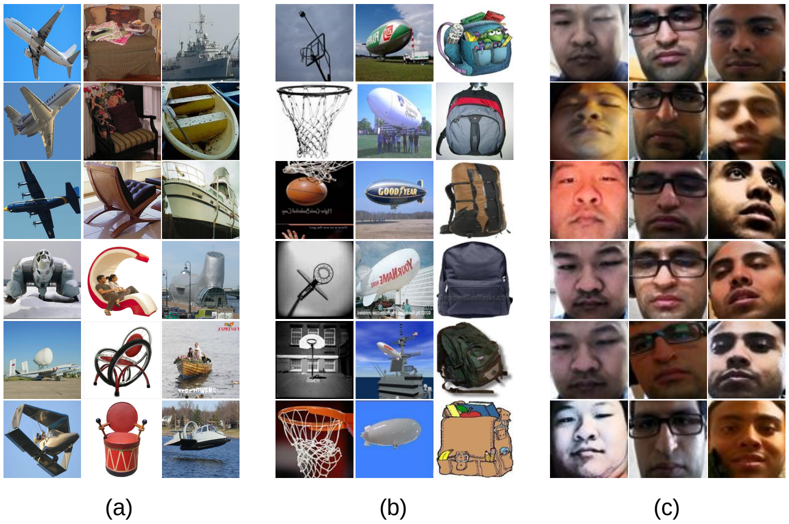
AlexNet Binary and VGG16 Binary (Baseline). A binary CNN is trained by having ImageNet samples and one-class image samples as the two classes using AlexNet and VGG16 architectures, respectively. Testing is performed using k-nearest neighbor, One-class SVM[20], Isolation Forest[13] and Gaussian Mixture Model[13] classifiers.

One-class Neural Network (OCNN). Method proposed in [71] applied on the extracted features from the AlexNet and VGG16 networks.

Autoencoder [15]. Network architecture proposed in [15] is used to learn a representation of the data. Reconstruction loss is used to perform verification.

Ours (AlexNet) and ours (VGG16). Proposed method applied with AlexNet and VGG16 network backbone architectures. The $fc7$ features are used during testing.

In addition to these baselines, in each experiment we report the performance of other task specific methods.



(a) (b) (c)

Figure 4-6. Sample images from datasets used for evaluation. (a)PASCAL VOC + Abnormal 1001 dataset. (b) Caltech 256 dataset. (c) UMDAA02 dataset.

Results

Abnormal Image Detection: The goal in abnormal image detection is to detect abnormal images when the classifier is trained using a set of normal images of the corresponding class. Since the nature of abnormality is unknown a priori, training is carried out using a single class (images belonging to the normal class). The 1001 Abnormal Objects Dataset [72] contains 1001 abnormal images belonging to six classes which are originally found in the PASCAL [73] dataset. Six classes considered in the dataset are Airplane, Boat, Car, Chair, Motorbike and Sofa. Each class has at least one hundred abnormal images in the dataset. A sample set of abnormal images and the corresponding normal images in the PASCAL dataset are show in Figure 4-6(a). Abnormality of images has been judged based on human responses received on the Amazon Mechanical Turk. We compare the performance of abnormal detection of the proposed framework with conventional CNN schemes and with the comparisons presented in [72]. It should be noted that our testing procedure is consistent with the protocol used in [72].

Results corresponding to this experiment are shown in Table 4-I. Adjusted graphical model presented in [72] has outperformed methods based on traditional deep features. The introduction of the proposed framework has improved the performance in AlexNet almost by a margin of 14%. Proposed method based on VGG produces the best performance on this dataset by introducing a 4.5% of an improvement as compared with the Adjusted Graphical Method proposed in [72].

Table 4-I. Abnormal image detection results on the 1001 Abnormal Objects dataset.

Method	AUC (Std. Dev.)
Graphical Model [72]	0.870
Adjusted Graphical Model [72]	0.911
Autoencoder[15]	0.674 (0.120)
OCNN AlexNet[71]	0.845 (0.148)
OCNN VGG16[71]	0.888 (0.0460)
AlexNet Features KNN	0.790 (0.074)
VGG16 Features KNN	0.847 (0.074)
AlexNet Binary KNN	0.621 (0.153)
VGG16 Binary KNN	0.848 (0.081)
AlexNet Features IF	0.613 (0.085)
VGG16 Features IF	0.731 (0.078)
AlexNet Binary IF	0.641 (0.089)
VGG16 Binary IF	0.715 (0.077)
AlexNet Features SVM	0.732 (0.094)
VGG16 Features SVM	0.847 (0.074)
AlexNet Binary SVM	0.736 (0.115)
VGG16 Binary SVM	0.834 (0.083)
AlexNet Features GMM	0.679 (0.103)
VGG16 Features GMM	0.818 (0.072)
AlexNet Binary GMM	0.696 (0.116)
VGG16 Binary GMM	0.803 (0.103)
DOC AlexNet (ours)	0.930 (0.032)
DOC VGG16 (ours)	**0.956 (0.031)**

One-Class Novelty Detection: In one-class novelty detection, the goal is to assess the novelty of a new sample based on previously observed samples. Since novel examples do not exist prior to test time, training is carried out using one-class learning principles. In the previous works [28],[29], the performance of novelty detection has been assessed based on different classes of the ImageNet and the Caltech 256 datasets. Since all CNNs used in our work have been trained using the ImageNet dataset, we use the Caltech 256 dataset to evaluate the performance of one-class novelty detection. The Caltech 256 dataset contains images belonging to 256 classes with total of 30607 images. In our experiments, each single class was considered separately and all other classes were considered as alien. Sample images belonging to three classes in the dataset are shown in Figure 4-6 (b). First, consistent with the protocol described in [29], AUC of 20 random repetitions were evaluated by considering the *American Flag* class as the known class and by considering boom-box, bulldozer and can-non classes as alien classes. Results corresponding to different methods are tabulated in Table 4-II.

In order to evaluate the robustness of our method, we carried out an additional test involving all classes of the Caltech 256 dataset. In this test, first a single class is chosen to be the enrolled class. Then, the effectiveness of the learned classifier was evaluated by considering samples from all other 255 classes. We did 40 iterations of the same experiment by considering first 40 classes of the Caltech 256 dataset one at a time as the enrolled class. Since there are 255 alien classes in this test as opposed to the first test, where there were only three alien classes, performance is expected to be lower than in the former. Results of this experiment are tabulated in Table 4-III.

It is evident from the results in Table 4-II that a significant improvement is obtained in the proposed method compared to previously proposed methods. However, as shown in Table 4-III this performance is not limited just to a American Flag. Approximately the same level of performance is seen across all classes in the Caltech 256 dataset. Proposed method has improved the performance of AlexNet by nearly 13% where as the improvement the proposed method has had on VGG16 is around 9%. It is interesting to note that binary CNN classifier based on the VGG framework has recorded performance very close to the proposed method in this task (difference in performance is about 1%). This is due to the fact that both ImageNet and Caltech 256 databases contain similar object classes. Therefore, in this particular case, ImageNet samples are a good representative of novel object classes present in Caltech 256. As a result of this special situation, binary CNN is able to produce results on par with the

Table 4-II. Novelty detection results on the Caltech 256 where *American Flag* class is taken as the known class.

Method	AUC (Std. Dev.)
One Class SVM [20]	0.606 (0.003)
KNFST [28]	0.575 (0.004)
Oc-KNFD [29]	0.619 (0.003)
Autoencoder[15]	0.532(0.003)
OCNN AlexNet[71]	0.907 (0.029)
OCNN VGG16[71]	0.943 (0.035)
AlexNet Features KNN	0.811 (0.003)
VGG16 Features KNN	0.951 (0.023)
AlexNet Binary KNN	0.920 (0.026)
VGG16 Binary KNN	0.997 (0.001)
AlexNet Features IF	0.836 (0.005)
VGG16 Features IF	0.910 (0.035)
AlexNet Binary IF	0.795 (0.007)
VGG16 Binary IF	0.907 (0.033)
AlexNet Features SVM	0.878 (0.007)
VGG16 Features SVM	0.951 (0.029)
AlexNet Binary SVM	0.920 (0.008)
VGG16 Binary SVM	0.942 (0.031)
AlexNet Features GMM	0.842 (0.004)
VGG16 Features GMM	0.901 (0.023)
AlexNet Binary GMM	0.860 (0.009)
VGG16 Binary GMM	0.924 (0.025)
DOC AlexNet (ours)	0.930 (0.005)
DOC VGG16 (ours)	**0.999 (0.001)**

proposed method. However, this result does not hold true in general as evident from other two experiments.

Table 4-III. Average Novelty detection results on the Caltech 256 dataset.

Method	AUC (Std. Dev.)
One Class SVM [20]	0.531 (0.120)
Autoencoder[15]	0.623 (0.072)
OCNN AlexNet[71]	0.826 (0.153)
OCNN VGG16[71]	0.885 (0.144)
AlexNet Features KNN	0.820 (0.062)
VGG16 Features KNN	0.897 (0.050)
AlexNet Binary KNN	0.860 (0.065)
VGG16 Binary KNN	0.902 (0.024)
AlexNet Features IF	0.794 (0.075)
VGG16 Features IF	0.890 (0.049)
AlexNet Binary IF	0.788 (0.087)
VGG16 Binary IF	0.891 (0.053)
AlexNet Features SVM	0.852 (0.057)
VGG16 Features SVM	0.902 (0.050)
AlexNet Binary SVM	0.856 (0.058)
VGG16 Binary SVM	0.909 (0.047)
AlexNet Features GMM	0.790 (0.083)
VGG16 Features GMM	0.852 (0.087)
AlexNet Binary GMM	0.801 (0.083)
VGG16 Binary GMM	0.870 (0.069)
DOC AlexNet (ours)	0.959 (0.021)
DOC VGG16 (ours)	**0.981 (0.022)**

Active Authentication (AA): In the final set of tests, we evaluate the performance of different methods on the UMDAA-02 mobile AA dataset [8]. The UMDAA-02 dataset contains multi-modal sensor observations captured over a span of two months from 48 users for the problem of continuous authentication. In this experiment, we only use the face images of users collected by the front-facing camera of the mobile device. The UMDAA-02 dataset is a highly challenging dataset with large amount of intra-class variation including pose, illumination and appearance variations. Sample images from the UMDAA-02 dataset are shown in Figure 4-6 (c). As a result of these high degrees of variations, in some cases the inter-class distance between different classes seem to be comparatively lower making recognition challenging.

During testing, we considered first 13 users taking one user at a time to represent the enrolled class where all the other users were considered to be alien. The performance of different methods on this dataset is tabulated in Table 4-IV.

Table 4-IV. Active Authentication results on the UMDAA-02 dataset.

Method	AUC (Std. Dev.)
One Class SVM [20]	0.594 (0.070)
Autoencoder[15]	0.643 (0.074)
OCNN AlexNet[71]	0.595 (0.045)
OCNN VGG16[71]	0.574 (0.039)
AlexNet Features KNN	0.708 (0.060)
VGG16 Features KNN	0.748 (0.082)
AlexNet Binary KNN	0.627 (0.128)
VGG16 Binary KNN	0.687 (0.086)
AlexNet Features IF	0.694 (0.075)
VGG16 Features IF	0.733 (0.080)
AlexNet Binary IF	0.625 (0.099)
VGG16 Binary IF	0.677 (0.078)
AlexNet Features SVM	0.702 (0.087)
VGG16 Features SVM	0.751 (0.075)
AlexNet Binary SVM	0.656 (0.112)
VGG16 Binary SVM	0.685 (0.076)
AlexNet Features GMM	0.690 (0.077)
VGG16 Features GMM	0.751 (0.082)
AlexNet Binary GMM	0.629 (0.110)
VGG16 Binary GMM	0.650 (0.087)
DOC AlexNet (ours)	0.786 (0.061)
DOC VGG16 (ours)	**0.810 (0.067)**

Recognition results are comparatively lower for this task compared to the other tasks considered in this paper. This is both due to the nature of the application and the dataset. However, similar to the other cases, there is a significant performance improvement in proposed method compared to the conventional CNN-based methods. In the case of AlexNet, improvement induced by the proposed method is nearly 8% whereas it is around 6% for VGG16. The best performance is obtained by the proposed method based on the VGG16 network.

Discussion

Analysis on mis-classifications: The proposed method produces better separation between the class under consideration and alien samples as presented in the results section. However, it is interesting to investigate on what conditions the proposed method fails. Shown in Figure 4-7 are a few cases where the proposed method produced erroneous detections for the problem of one-class novelty detection with respect to the *American Flag* class (in this experiment, all other classes in Caltech256 dataset were used as alien classes). Here, detection threshold has been selected as $\delta = 0$. Mean detection scores for *American Flag* and alien images were 0.0398 and 8.8884, respectively.

As can be see from Figure 4-7, in majority of false negative cases, the American Flag either appears in the background of the image or it is too closer to clearly identify its characteristics. On the other hand, false positive images either predominantly have American flag colors or the texture of a waving flag. It should be noted that the nature of mis-classifications obtained in this experiment are very similar to that of multi-class CNN-based classification.

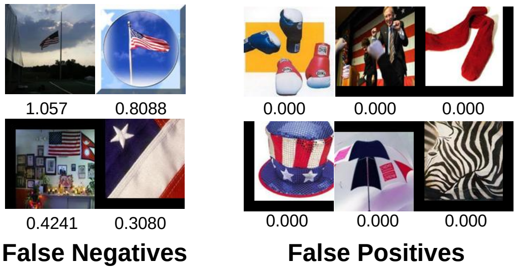


Figure 4-7. Sample false detections for the one-class problem of novelty detection (*American Flag*).

Using a subset of the reference dataset: In practice, the reference dataset is often enormous in size. For example, the ImageNet dataset has in excess of one million images. Therefore, using the whole reference dataset for transfer learning may be inconvenient. Due to the low number of training iterations required, it is possible to use a subset of the original reference dataset in place of the reference dataset without causing over-fitting. In our experiments, training of the reference network was done using the validation set of the ImageNet dataset. Recall that initially, both networks are loaded with pre-trained models. It should be noted that these pre-trained models have to be trained using the whole reference dataset. Otherwise, the resulting network will have poor generalization properties.

Number of training iterations: In an event when only a subset of the original reference dataset is used, the training process should be closely monitored. It is best if training can be terminated as soon as the composite loss converges. Training the network long after composite loss has converged could result in inferior features due to over-fitting. This is the trade-off of using a subset of the reference dataset. In our experiments, convergence occurred around 2 epochs for all test cases (Figure 4-5). We used a fixed number of iterations (700) for each dataset in our experiments.

Effect of number of templates: In all conducted experiments, we fixed the number of templates used for recognition to 40. In order to analyze the effect of template size on the performance of our method, we conducted an experiment by varying the template size. We considered two cases: first, the novelty detection problem related to the *American Flag* (all other classes in Caltech256 dataset were used as alien classes), where the recognition rates were very high at 98%; secondly, the AA problem where the recognition results were modest. We considered Ph01USER002 from the UMDAA-02 dataset for the study on AA. We carried out twenty iterations of testing for each case. The obtained results are tabulated in Table 4-V.

According to the results in Table 4-V, it appears that when the proposed method is able to isolate a class sufficiently, as in the case of novelty detection, the choice

Table 4-V. Mean AUC (with standard deviation values in brackets) obtained for different template sizes.

Number of templates	1	5	10	20	30	40
American Flag (Caltech 256)	0.987 (0.0034)	**0.988** (0.0032)	0.987 (0.0030)	0.988 (0.0038)	0.988 (0.0029)	0.988 (0.0045)
Ph01USER002 (UMDAA02)	0.762 (0.0226)	0.788 (0.0270)	0.806 (0.0134)	0.787 (0.0262)	0.821 (0.0165)	**0.823** (0.0168)

of the number of templates is not important. Note that even a single template can generate significantly accurate results. However, this is not the case for AA. Reported relatively lower AUC values in testing suggests that all faces of different users lie in a smaller subspace. In such a scenario, using more templates have generated better AUC values.

Impact of Different Features

In this subsect, we investigate the impact of different choices of h_c and g has on the recognition performance. Feature was varied from $fc6$ to $fc8$ and the performance of the abnormality detection task was evaluated. When $fc6$ was used as the feature, the sub-network g consisted layers from conv1 to $fc6$, where layers $fc7$ and $fc8$ were associated with the sub network h_c. Similarly, when the layer $fc7$ was considered as the feature, the sub-networks g and h_c consisted of layers conv1 $- fc7$ and $fc8$, respectively.

In Table 4-VI, the recognition performance on abnormality image detection task is tabulated for different choices of h_c and g. From Table 4-VI we see that in both AlexNet and VGG16 architectures, extracting features at a later layer has yielded in better performance in general. For example, for VGG16 extracting features from $fc6, fc7$ and $fc8$ layers has yielded AUC of $0.856, 0.956$ and 0.969, respectively. This observation is not surprising on two accounts. First, it is well-known that later layers of deep networks result in better generalization. Secondly, *Compactness Loss* is minimized with respect to features of the target dataset extracted in the $fc8$ layer.

Therefore, it is expected that $fc8$ layer provides better compactness in the target dataset.

Table 4-VI. Abnormal image detection results for different choices of the reference dataset.

	fc6	fc7	fc8
DOC AlexNet (ours)	0.936 (0.041)	0.930 (0.032)	**0.947 (0.035)**
DOC VGG16 (ours)	0.856 (0.118)	0.956 (0.031)	**0.969 (0.029)**

Impact of the Reference Dataset

The proposed method utilizes a reference dataset to ensure that the learned feature is informative by minimizing the *descriptiveness loss*. For this scheme to result in effective features, the reference dataset has to be a non-trivial multi-class object dataset. In this subsect, we investigate the impact of the reference dataset on the recognition performance. In particular, abnormal image detection experiment on the Abnormal 1001 dataset was repeated with a different choice of the reference dataset. In this experiment ILSVRC12 [63], Places365 [64] and Oxford Flowers 102 [74] datasets were used as the reference dataset. We used publicly available pre-trained networks from caffe model zoo [70] in our evaluations.

In Table 4-VII the recognition performance for the proposed method as well as the baseline methods are tabulated for each considered dataset. From Table 4-VII we observe that the recognition performance has dropped when a different reference dataset is used in the case of VGG16 architecture. Places365 has resulted in a drop of 0.038 whereas the Oxford flowers 102 dataset has resulted in a drop of 0.026. When the AlexNet architecture is used, a similar trend can be observed. Since Places365 has smaller number of classes than ILVRC12, it is reasonable to assume that the latter is more diverse in content. As a result, it has helped the network to learn more informative features. On the other hand, although Oxford flowers 102 has even fewer classes, it should be noted that it is a fine-grain classification dataset. As a result, it too has helped to learn more informative features compared to Places365. However, due to the presence of large number of non-trivial classes, the ILVRC12 dataset has yielded the best performance among the considered cases.

Table 4-VII. Abnormal image detection results for different choices of the reference dataset.

	ILVRC12	Places365	Flowers102
AlexNet Features KNN	0.790 (0.074)	0.856 (0.056)	0.819 (0.075)
VGG16 Features KNN	0.847 (0.074)	0.809 (0.100)	0.828 (0.077)
AlexNet Binary KNN	0.621 (0.153)	0.851 (0.060)	0.823 (0.084)
VGG16 Binary KNN	0.848 (0.081)	0.837 (0.090)	0.839 (0.077)
AlexNet Features IF	0.613 (0.085)	0.771 (0.107)	0.739 (0.098)
VGG16 Features IF	0.731 (0.078)	0.595 (0.179)	0.685 (0.154)
AlexNet Binary IF	0.641 (0.089)	0.777 (0.092)	0.699 (0.129)
VGG16 Binary IF	0.715 (0.077)	0.637 (0.159)	0.777 (0.110)
AlexNet Features SVM	0.732 (0.094)	0.839 (0.062)	0.818 (0.076)
VGG16 Features SVM	0.847 (0.074)	0.776 (0.113)	0.826 (0.077)
AlexNet Binary SVM	0.736 (0.115)	0.847 (0.065)	0.823 (0.083)
VGG16 Binary SVM	0.834 (0.083)	0.789 (0.114)	0.788 (0.089)
AlexNet Features GMM	0.679 (0.103)	0.832 (0.069)	0.779 (0.076)
VGG16 Features GMM	0.818 (0.072)	0.782 (0.103)	0.771 (0.114)
AlexNet Binary GMM	0.696 (0.116)	0.835 (0.068)	0.815 (0.101)
VGG16 Binary GMM	0.803 (0.103)	0.770 (0.103)	0.777 (0.110)
DOC AlexNet (ours)	0.930 (0.032)	0.896 (0.019)	0.899 (0.052)
DOC VGG16 (ours)	**0.956 (0.031)**	**0.918 (0.049)**	**0.930 (0.059)**

Derivations

Batch-variance Loss is a Scaled Version of Sample Variance: Consider the definition of batch-variance loss defined as,

$l_b = \frac{1}{nk}\sum_{i=1}^{n} \mathbf{z_i}^T\mathbf{z_i}$ where, $\mathbf{z_i} = \left[\mathbf{x_i} - \frac{1}{n-1}\sum_{j\neq i}\mathbf{x_j}\right]$. Re-arranging terms in $\mathbf{z_i}$,

$$\mathbf{z_i} = \left[\mathbf{x_i} - \frac{1}{n-1}\sum_{j=1}^{n}\mathbf{x_j} + \frac{1}{n-1}\mathbf{x_i}\right]$$

$$\mathbf{z_i} = \left[\frac{n}{n-1}\mathbf{x_i} - \frac{1}{n-1}\sum_{j=1}^{n}\mathbf{x_j}\right]$$

$$\mathbf{z_i} = \frac{n}{n-1}\left[\mathbf{x_i} - \frac{1}{n}\sum_{j=1}^{n}\mathbf{x_j}\right]$$

$$\mathbf{z_i}^T\mathbf{z_i} = \frac{n^2}{(n-1)^2}\left[\mathbf{x_i} - \frac{1}{n}\sum_{j=1}^{n}\mathbf{x_j}\right]^T\left[\mathbf{x_i} - \frac{1}{n}\sum_{j=1}^{n}\mathbf{x_j}\right]$$

But, $\left[\mathbf{x_i} - \frac{1}{n}\sum_{j=1}^{n}\mathbf{x_j}\right]^T\left[\mathbf{x_i} - \frac{1}{n}\sum_{j=1}^{n}\mathbf{x_j}\right]$ is the sample variance σ_i^2. Therefore,

$$l_b = \frac{1}{nk}\sum_{i=1}^{n}\frac{n^2\sigma_i^2}{(n-1)^2}$$

Therefore, $l_b = \beta\sigma^2$, where $\beta = \frac{n^2}{k(n-1)^2}$ is a constant and σ^2 is the average sample variance.

Backpropagation of Batch-variance Loss:

Consider the definition of batch variance loss l_b,

$l_b = \frac{1}{nk}\sum_{i=1}^{n}\mathbf{z_i}^T\mathbf{z_i}$ where, $\mathbf{z_i} = \mathbf{x_i} - \mathbf{m_i}$.

From the definition of the inner product,

$\mathbf{z_i}^{\mathbf{T}}\mathbf{z_i} = \sum_{j=1}^{k} z_{ij}{}^2$. Therefore, l_b can be written as,

$$l_b = \frac{1}{nk}\sum_{i=1}^{n}\sum_{l=1}^{k}(x_{il} - m_{il})^2.$$

Taking partial derivatives of l_b with respect to x_{ij}. By chain rule we obtain,

$$\frac{\partial l_b}{\partial x_{ij}} = \frac{2}{nk}\sum_{l=1}^{k} x_{il} - m_{il} \times \frac{\partial x_{il} - m_{il}}{\partial x_{ij}}.$$

Note that $\frac{\partial x_{ij} - m_{ij}}{\partial x_{ij}} = 1$ when $j = l$. Otherwise, $\frac{\partial x_{ij} - m_{ij}}{\partial x_{ij}} = -\frac{\partial m_{ij}}{\partial x_{ij}} = \frac{-1}{n-1}$.

$$\frac{\partial l_b}{\partial x_{ij}} = \frac{2}{nk}\left[x_{ij} - m_{ij} - \frac{1}{n-1}\sum_{l \neq j} x_{il} - m_{il}\right].$$

$$\frac{\partial l_b}{\partial x_{ij}} = \frac{2}{nk}\left[\frac{n}{n-1}x_{ij} - m_{ij} - \frac{1}{n-1}\sum_{l=1}^{n} x_{il} - m_{il}\right].$$

$$\frac{\partial l_b}{\partial x_{ij}} = \frac{2}{(n-1)nk}\left[n \times (x_{ij} - m_{ij}) - \sum_{l=1}^{n}(x_{il} - m_{il})\right].$$

Detailed Network Architectures

Shown in Figure 4-8 are the adaptations of the proposed method to the existing CNN architectures- (a) AlexNet and (b) VGG16, respectively. We have used these two architectures to carry out all experiments in this paper. In both architectures, two images from the target dataset t and the reference dataset r are fed into the network to evaluate cross entropy loss and batch-variance loss. Weights of convolution and fully-connected layers are shared between the two branches of the network. For all experiments, stochastic gradient descent algorithm is used with a learning rate of 5×10^{-5} and a weight decay of 0.0005.

Summary

We introduced a deep learning solution for the problem of one-class classification, where training samples of a single class are available along with out-of-distribution data during training. We proposed a feature learning scheme that engineers class-specific features that are generically discriminative. To facilitate the learning process, we proposed two loss functions *descriptiveness loss* and *compactness loss* with a CNN network structure. Proposed network structure could be based on any CNN backbone of choice. The effectiveness of the proposed method is shown in results for AlexNet and VGG16-based backbone architectures. The performance of the proposed method is tested on publicly available datasets for abnormal image detection, novelty detection and face-based mobile active authentication. The proposed method obtained the state-of-the-art performance in each test case.

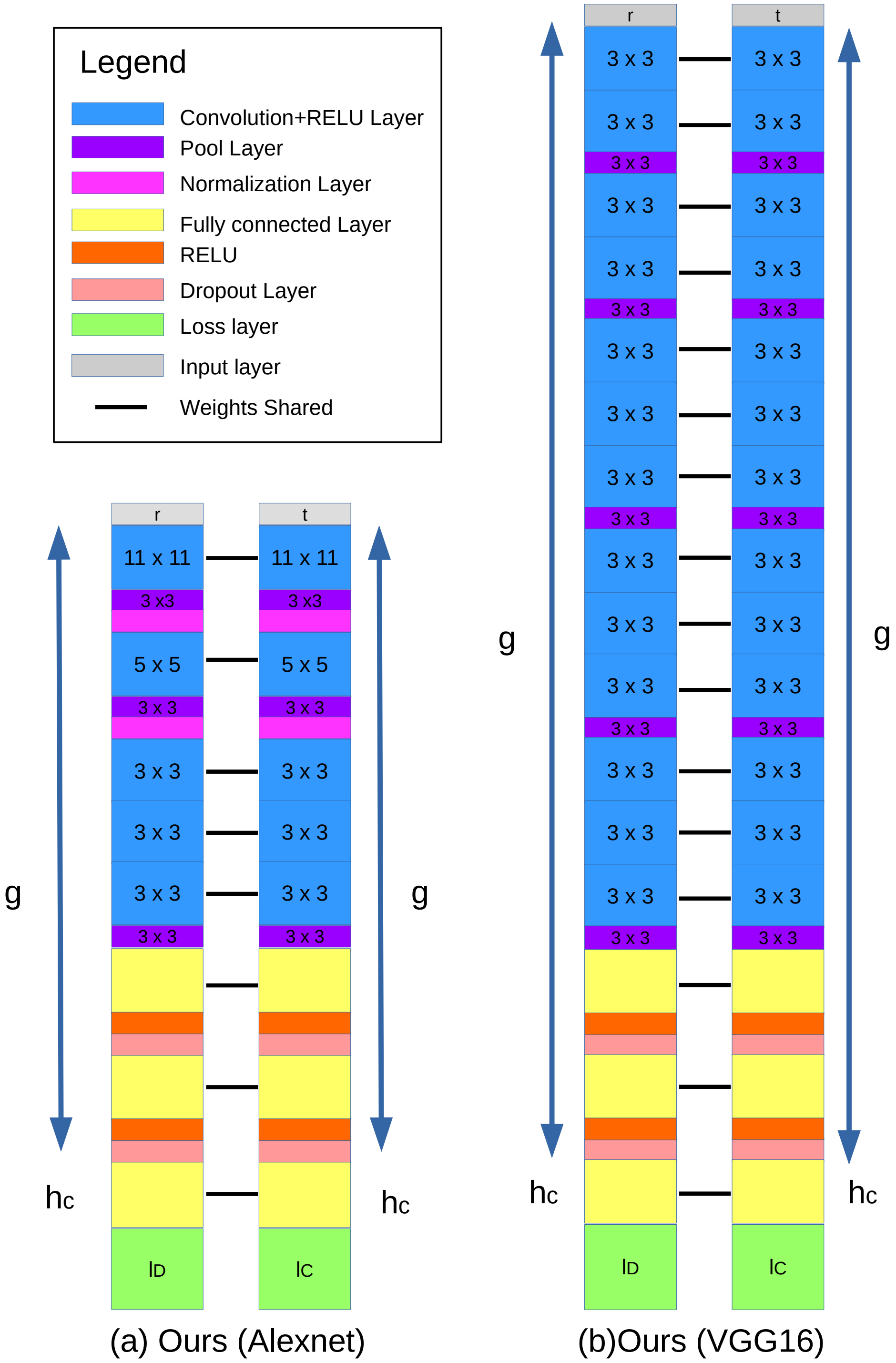


Figure 4-8. CNN architectures based on AlexNet and VGG16 backbones for the proposed method.

Chapter 5

One-class Novelty Detection

As discussed earlier, with the advent of deep learning, one-class novelty detection has received considerable amount of attention in the literature. Contemporary works in one-class novelty detection focus on learning a representative latent space for the given class [16, 22]. Once such a space is learned, novelty detection is performed based on the projection of a query image onto the learned latent space. Two distinct strategies are commonly used for this purpose in the literature. In the first strategy, the difference between the query image and its inverse image (reconstruction) is used as a novelty detector. Various distance measures ranging from mean squared error [16] to discriminator output [17] have been used in the literature for this purpose. In comparison, the second strategy explicitly models the learned latent space using a distribution [12, 22, 23]. We consider the former strategy for novelty detection. We investigate limitations of existing representation learning techniques and propose learning a latent space that *exclusively* generates only in-class examples, to improve performance in novelty detection.

Existing work focuses on generating a latent representation that preserves details of the given class. In doing so, it is assumed that when an out-of-class object is presented to the network, it will do a poor job of describing the object, thereby reporting a relatively higher reconstruction error. However, this assumption does not hold at all times. For example, experiments done on digits in the literature [22, 23] suggest that

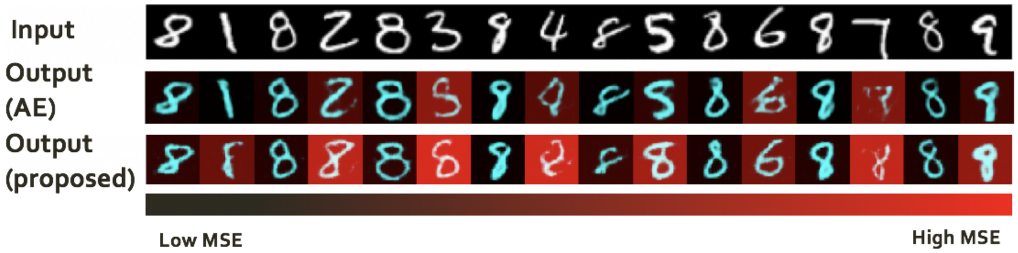


Figure 5-1. Limitations of in-class representation based novelty detection.

networks such as auto-encoders trained on digits with a simple shape such as 0 and 1 have high novelty detection accuracy. In contrast, digits with complex shapes, such as digit 8, have relatively weaker novelty detection accuracy. This is because a latent space learned for a class with complex shapes inherently learns to represent some of out-of-class objects as well. As an example, the latent space learned on digit 8 is also able to represent other digits such as 1,3,6,7 reasonably well – thereby producing very low distance error values for out-of-class examples as shown in Figure 5-1 (middle). We note that the requirement in novelty detection is not only to ensure that in-class samples are well represented; it is also to ensure that out-of-class samples are poorly represented. To the best of our knowledge, none of the previous work has addressed the latter requirement. In this work, we propose One-Class GAN (OCGAN), a two-fold latent space learning process that considers both these requirements.

At a high-level, we learn a latent space that represents objects of a given class well. Secondly, we ensure that any example generated from the learned latent space is indeed from the known class. In other words, if the network is trained on a digits of 8, we ensure that any sample drawn from the latent space, when used to generate an image, corresponds to an image of digit 8. This ensures that out-of-class samples are not well represented by the network. Shown in Figure 5-1(bottom) are the outputs generated by the proposed method for the inputs shown in Figure 5-1(top). Since the entire latent space corresponds to images from digit 8, all projections into the latent space in return produce images of digit 8

Motivation

Let us reconsider an autoencoder trained on digit 8 where a network trained to represent a given class has ended up providing good representation for images of other classes. When images of a given class are sufficiently diverse, smoothly transitioning between the projection of one in-class image in the latent space to that of another can be done along infinitely many different paths – this is particularly the case for latent spaces with high dimensionality. In training auto-encoders, we model projections of only observed examples into the latent space - not all possible paths between the corresponding latent points.

In Figure 5-2 we visualize a path traced in the latent space between two points corresponding to two different images of the given class (class 8). This visualization reveals that as we transition from one point to the other in the latent space along the specified path, certain intermediate latent samples resemble the likeness of digit 1. When the network observes an instance of digit 1, it gets projected onto such samples. Since digit 1 is well represented by the network, its reconstruction error will be low, although it is out of class. The core idea of our proposal is based on this observation. We argue that if the entire latent space is constrained to represent images of the given class, the representation of out-of-class samples will be minimal – thereby producing high reconstruction errors for them.

With this strategy in mind, we explicitly force the entirety of the latent space to represent only the given class. When applied to the example in Fig. 5-2, all latent samples along any path between the two 8's will reconstruct into a set of digit 8 images. Visualization of the path as shown in Figure 5-2(b) validates this claim. As a result, when an out-of-class digit 1 is presented to the model, there will be a high difference between the digit and the reconstruction of the digit (which will now look more like a digit 8). As a result, the proposed method is able to produce superior

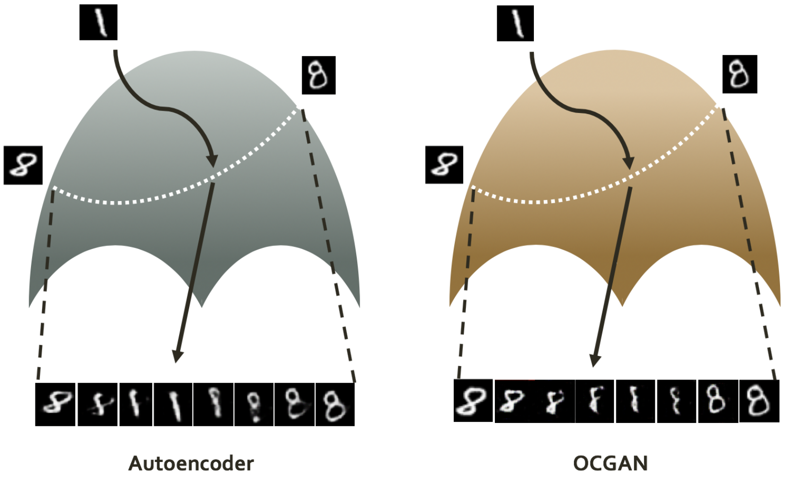


Figure 5-2. Illustration of the latent space learned for digit 8 using a denoising-autoencoder network (left) and OCGAN(right).

novelty detection performance.

Proposed Strategy

The proposed solution, OCGAN, consists of four components: a denoising auto-encoder, two discriminators (latent and visual discriminator) and a classifier. The proposed network is trained using adversarial principles. We describe each of these components in detail below.

Denoising auto-encoder: Following previous work, we use a denoising auto-encoder network to learn a representation for the given concept. Encoder network and the Decoder network are denoted by symbols En and De respectively. Our strategy revolves around densely sampling from the latent space. To facilitate this operation, with the intention of having a bounded support for the latent space, we introduce a *tanh* activation in the output layer of the encoder. Therefore, support of the latent space is $(-1, 1)^d$, where d is the dimension of the latent space. In our implementation, we add zero mean Gaussian white noise with a variance of 0.2 to input images and train the auto-encoder using mean squared error loss as shown below:

$$l_{\text{MSE}} = \|x - \text{De}(\text{En}(x+n))\|_2^2, \tag{5.1}$$

where x is an input image and $n \sim \mathcal{N}(0, 0.2)$. In addition, adversarial loss terms introduced in the following sections are also used to learn parameters of the auto-encoder. Since the decoder part of the auto-encoder also acts as the generator of images from latent space, we use the words *decoder* and *generator* interchangeably in the remainder of the text.

Latent Discriminator: The motivation of our method is to obtain a latent space where each and every instance from the latent space represents an image from the given class. If representations of the given class are only confined to a sub-region of the latent space, this goal is not possible to achieve. Therefore, we explicitly force latent representations of in-class examples to be distributed uniformly across the latent space. We achieve this using a discriminator operating in the latent space that we call *latent discriminator* D_l . The latent discriminator is trained to differentiate between latent representations of real images of the given class and samples drawn from a $\mathbb{U}(-1,1)^d$ distribution. We consider a loss of the form:

$$l_{\text{latent}} = -(\mathbb{E}_{s\sim\mathbb{U}(-1,1)}[\log D_l(s)] + \mathbb{E}_{x\sim p_x}[\log(1 - D_l(\text{En}(x+n)))]) \tag{5.2}$$

where, p_x is the distribution of in-class examples. We train the latent discriminator along with the auto-encoder network using $\max_{\text{En}} \min_{D_l} l_{\text{latent}}$. Since the latent space is a hyper-cube with support $(-1,1)^d$, at equilibrium, the latent projections of examples from the given class are expected to be distributed evenly following a $\mathbb{U}(-1,1)^d$ distribution.

Visual Discriminator: In order for the network not to represent any out-of-class objects, we propose to sample exhaustively from the latent space and ensure corresponding images are not from out-of class. Since there are no negative classes present during training, this condition is difficult to enforce. Instead, we make sure that all images generated from latent samples are from the same image space distribution as the given class. In order to enforce this constraint, we use a second discriminator that we call *visual* discriminator (D_v).

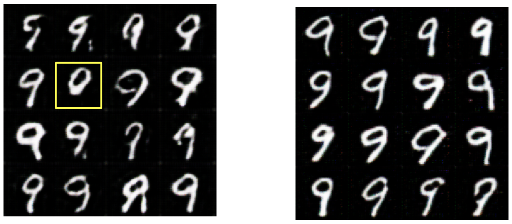
Visual discriminator is trained to differentiate between images of the given class and images generated from random latent samples using the decoder $\mathrm{De}(s)$, where s is a random latent sample. We refer to latter images as *fake images* for the remainder of the paper. When the visual discriminator is fooled, fake images chosen at random in general will look similar to examples from the given class. We evaluate adversarial loss l_{visual} as follows.

$$l_{\text{visual}} = -(\mathbb{E}_{s\sim\mathbb{U}(-1,1)}[\log D_v(\mathrm{De}(s))] + \mathbb{E}_{x\sim p_l}[\log(1 - D_v(x))]). \qquad (5.3)$$

We learn visual discriminator together with the auto-encoder network using $\max_{\mathrm{De}} \min_{D_v} l_{\text{visual}}$.

Informative-negative Mining: The components described thus far account for the core of the proposed network. Shown in Figure 5-3(a) is a visualization of fake images obtained by jointly training these three sub-networks using digit 9. Figure 5-3(a) suggests that the proposed network is able to generate plausible images of the given class for majority of the random latent samples. However, as indicated in the figure there are few cases where the produced output looks different from the given class. For example, the highlighted digit in Figure 5-3(a) looks more like a zero than a nine.

This result suggests that despite the proposed training procedure, there are latent space regions that do not produce images of the given class. This is because sampling from all regions in the latent space is impossible during training – particularly when the latent dimension is large. A naive solution to this problem is to reduce the dimensionality of the latent space. However, with a lower dimension, the amount of detail the network preserves goes down. As a result, although all latent samples



(a) (b)

Figure 5-3. Visualization of generated images from random latent samples when the network is trained (a) without and (b) with informative-negative mining.

produce an in-class image, a very low dimensionality would diminish performance in novelty detection.

As an alternative, we propose to actively seek regions in the latent space that produce images of poor quality. For the remainder of the paper we refer to these images as *informative-negative* samples. We use informative-negative samples to train the generator so that it learns to produce good quality in-class images even for these latent samples. However, we continue to use samples chosen at random to train two discriminators, as feeding weaker samples would hinder training of discriminators. In order to find informative-negative samples, first we start with random latent-space samples and use a classifier to assess the quality of the image generated from the sample. The loss of the classifier is used to back-propagate and compute gradients in the latent space. We then take a small step in the direction of the gradient to move to a new point in the latent space where the classifier is confident that the generated image is out-of-class.

Classifier: The role of the classifier is to determine how well the given image resembles content of the given class. Ideally such a classifier can be trained using positive and negative examples of a given class. However, since there are no negative training samples available, we train a weaker classifier instead. In the proposed mechanism, if the content belongs to the given class, the classifier deems it positive, and if the content bears no resemblances to the positive class, the classifier deems it negative.

We train the classifier using reconstructions of in-class samples as positives and fake images, those that are generated from random samples in the latent space, as negatives. This classifier is trained independent of other network elements using binary cross entropy loss $l_{\text{classifier}}$. In other words, the classifier loss is not considered while learning generator and discriminator parameters. Initially, since the quality of fake samples is poor, the classifier is able to obtain very low loss value. As the quality of fake images improves with training, differentiation becomes harder and it forces the classifier to become smarter.

It should be noted that the classifier's prediction of a given image as a negative may or may not mean that the given image always corresponds to an informative-negative latent sample. Even if it does not, such images do not hinder the training process at all, and training proceeds as usual.

Since the informative-negative classifier does not participate in the GAN game, there is no requirement to balance the capacity of the classifier with the generator (whereas, this is the case for both other discriminators). Therefore, it is possible to make the classifier very strong to increase its confidence in in-class reconstructions.

Figure 5-4 shows the impact of the informative-negative mining procedure using a few illustrative examples. In the figure, image pairs before and after negative mining are displayed. We have shown cases where the original images are not largely changed in the bottom row. In the top row we have shown a few examples where the input images have been substantially altered as a result of informative-negative mining. For example, the top left sample of digit 2 appears to be a digit 7 after the process. In Figure 5-3(b), we show the impact of this procedure by visualizing a few fake images generated from random latent samples for digit 9. It is evident from the figure that informative-negative mining has helped in generating digits of the desired class more consistently across the whole latent space.

Full OCGAN Model: The full network of OCGAN and the breakdown of each

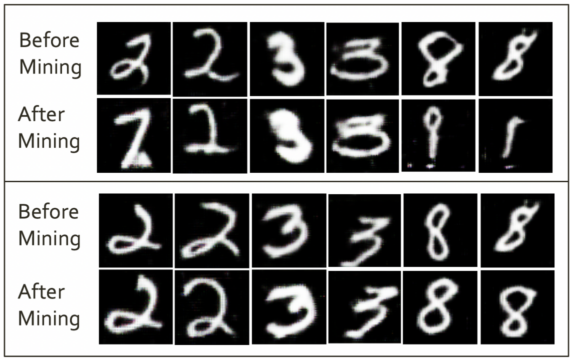


Figure 5-4. Effectiveness of informative-negative mining.

individual component of the proposed network is shown in Figure 5-5. The network is trained in two iterative steps. In the first step, all other sub-networks except the classifier network are frozen. The classifier network is trained with reconstructed in-class examples and generated fake examples. Then, it is frozen and the auto-encoder and two discriminators are trained adversarially. The latent discriminator is trained based on latent projections of in-class images and random samples drawn from $\mathbb{U}(-1, 1)$ distribution. The visual discriminator is trained using fake images generated from random latent samples and real images from the given class. Discriminators are trained by minimizing the loss $l_{\text{latent}} + l_{\text{visual}}$.

Prior to each generator step, informative-negative samples are sought in the latent space using a batch of random samples drawn from the $\mathbb{U}(-1, 1)$ distribution, and using gradient descent steps from the classifier's loss in the latent space. The auto-encoder is trained using informative-negative samples and latent projections of (noise-injected) real examples of the given class using $10 \times l_{\text{MSE}} + l_{\text{visual}} + l_{\text{latent}}$. A larger weight is given to the l_{MSE} term to obtain good reconstructions. The coefficient was chosen empirically based on the quality of reconstruction. In our implementation, we started mining for informative-negative samples only after the network started producing fake images of reasonable quality. Steps of the training procedure is summarized in Algorithm 1.

```
Input    : Set of training data $x$, iteration size $n$
Output : Models: En, De
for iteration 1 to $\rightarrow n$ do
    Classifier update: keep $D_l$, $D_v$, En, De fixed.
    $l_1 = \mathrm{En}(x + n)$
    $l_2 = \mathbb{U}(-1, 1)$
    $l_{\text{classifier}} = C(\mathrm{De}(l_2), 0) + C(\mathrm{De}(l_1), 1)$
    Back-propagate $l_{\text{classifier}}$ to change $C$

    Discriminator update:
    $l_{\text{latent}} = D_l(l_1, 0) + D_l(l_2, 1)$
    $l_{\text{visual}} = D_v(\mathrm{De}(l_2), 0) + D_v(x, 1)$
    Back-propagate $l_{\text{latent}} + l_{\text{visual}}$ and change $D_l, D_v$

    Negative mining : Keep all networks fixed.
    for sub-iteration 1 to $\rightarrow 5$ do
        $l_{\text{classifier}} = C(\mathrm{De}(l_2), 1)$
        Back-propagate  $l_{\text{classifier}}$ to change $l_2$
    end

    Generator update: keep $D_l, D_v$, C fixed.
    $l_{\text{latent}} = D_l(l_1, 1) + D_l(l_2, 0)$
    $l_{\text{visual}} = D_v(\mathrm{De}(l_2), 1) + D_v(x, 0)$
    $l_{\text{mse}} = ||x - \mathrm{De}(l_1)||^2$
    Back-propagate $l_{\text{latent}} + l_{\text{visual}} + \lambda l_{\text{mse}}$ to change En, De
end
```

Algorithm 1: Training methodology of the OCGAN model: D_l, D_v and C represent the outputs of the latent discriminator, visual discriminator and the classifier respectively. *En* and *De* are the encoder and the decoder/generator respectively. Real label and fake label are denoted by 1 and 0 respectively.

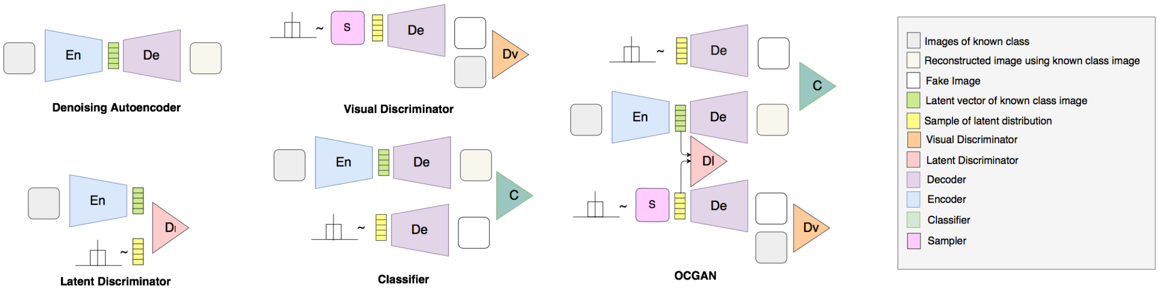


Figure 5-5. Illustration of OCGAN architecture: the network consists of four sub-networks : an auto-encoder, two discriminators and a classifier.

Network Architecture and Hyper-parameter Selection: The auto-encoder is a symmetric network with three 5 x 5 convolutions with stride 2 followed by three transposed convolutions. All convolutions and transposed-convolutions are followed by batch normalization and leaky $ReLU$ (with slope 0.2) operations. A $tanh$ activation was placed immediately after the last convolution layer to restrict support of the latent dimension. We used a base channel size of 64 for the auto-encoder and increased number of channels by a factor of 2 with every layer.

The visual discriminator and classifier are networks with three 5 x 5 convolutions with stride 2. Base channel size of two networks were chosen to be 12 and 64 respectively. Latent discriminator is a fully connected network with layers of sizes 128, 64, 32 and 16 respectively. Batch normalization and $ReLu$ activations were used after each layer in all networks.

At the end of training, we selected the model that resulted in minimum MSE on the validation set for evaluation. Model hyper-parameters such as learning rate, latent space size were chosen based on the MSE of validation set. The number of base channels in each network and coefficient of loss terms were decided based on the plot of training loss of each network component.

Experimental Results

Evaluation Methodology

We test the effectiveness of the proposed method using four publicly available multi-class object recognition datasets. In order to simulate a one-class setting, each class at a time is considered as the known class, as proposed in [23], [22] and [12]. The network is trained using only samples of the known class. During testing, we treat the union of remaining classes as out-of-class samples. Following previous work, we compare the performance of our method using Area Under the Curve (AUC) of Receiver Operating Characteristics (ROC) curve. Here, we note that there exist two protocols in the literature for one-class novelty detection.

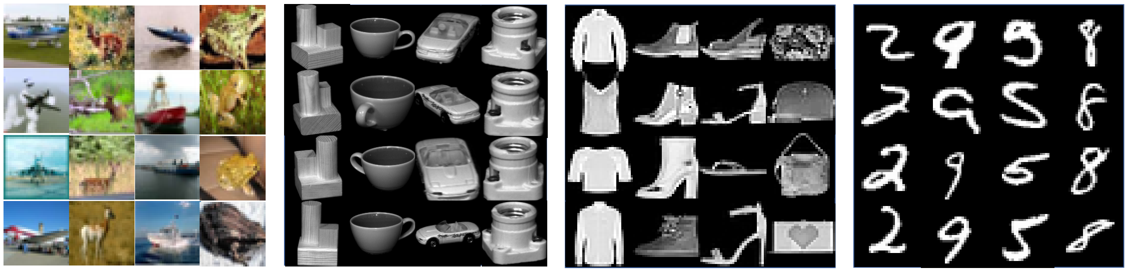
Protocol 1 : Training is carried out using 80% of in-class samples. The remaining 20% of in-class data is used for testing. Negative test samples are randomly selected so that they constitute half of the test set.

Protocol 2 : Use the training-testing splits of the given dataset to conduct training. Training split of the known class is used for training / validation. Testing data of all classes are used for testing.

The work of [22] used the 2^{nd} protocol to evaluate their performance in MNIST[75], FMNIST[76] and COIL100[77] datasets, whereas the authors of [23] and [12] chose the 1^{st} protocol on MNIST and CIFAR10[65] datasets. We compare our method on these baselines using the relevant protocol for fair comparison.

Datasets and Experimental Results

In this section we briefly introduce each dataset used for evaluation and present experimental results for the proposed method. In Figure 5-6, a few representative examples from the considered datasets are shown. We tabulate results corresponding to Protocol 1 in Table 5-I and results of protocol 2 in Tables 5-II and 5-III.



CIFAR10 COIL FMNIST MNIST

Figure 5-6. Representative images from the datasets used for evaluation. Images in each column belong to the same class.

COIL100 : COIL100 is a multi-class dataset where each object class is captured using multiple different poses. There are 100 image classes in the dataset with a few images per class (typically less than hundred). Figure 5-6 suggests that the intra-class difference is very small for this dataset. As a result, all considered method produces high AUC values for protocol 1 as shown in Table 5-I. Our proposed method of OCGAN records 0.995 AUC, surpassing [22] which reported AUC of 0.968.

fMNIST : fMNIST is intended to be a replacement for MNIST, where the dataset comprises of 28×28 images of fashion apparels/accessories. As evident from Figure 5-6, fMNIST is a more challenging dataset compared to both COIL100 and MNIST, since there is considerable amount of intra-class variances. The proposed method improves novelty detection performance by over 2% compared to [22] for this dataset, using protocol 1.

MNIST : MNIST dataset contains hand-written digits from 0-9 with a 28×28 resolution. This dataset has been widely used to benchmark one-class novelty detection results. In terms of complexity, it is an easier dataset compared to fMNIST, but more challenging than COIL100. We report performances of the proposed method on this dataset using both protocols.

When protocol 1 was used, our OCGAN model yielded an improvement of about 3% compared to state-of-the-art [22] method. As shown in Table 5-II, when protocol

Table 5-I. Mean One-class novelty detection using Protocol 1.

	MNIST	COIL	fMNIST
ALOCC DR [17]	0.88	0.809	0.753
ALOCC D [17]	0.82	0.686	0.601
DCAE [16]	0.899	0.949	0.908
GPND [22]	0.932	0.968	0.901
Ours: OCGAN	**0.977**	**0.995**	**0.924**

Table 5-II. One-class novelty detection results for MNIST dataset using Protocol 2.

	0	1	2	3	4	5	6	7	8	9	MEAN
OCSVM [20]	0.988	**0.999**	0.902	0.950	0.955	0.968	0.978	0.965	0.853	0.955	0.9513
KDE [13]	0.885	0.996	0.710	0.693	0.844	0.776	0.861	0.884	0.669	0.825	0.8143
DAE [15]	0.894	**0.999**	0.792	0.851	0.888	0.819	0.944	0.922	0.740	0.917	0.8766
VAE [38]	0.997	**0.999**	0.936	0.959	0.973	0.964	0.993	0.976	0.923	0.976	0.9696
Pix CNN [78]	0.531	0.995	0.476	0.517	0.739	0.542	0.592	0.789	0.340	0.662	0.6183
GAN [19]	0.926	0.995	0.805	0.818	0.823	0.803	0.890	0.898	0.817	0.887	0.8662
AND [23]	0.984	0.995	**0.947**	0.952	0.960	0.971	**0.991**	0.970	0.922	0.979	0.9671
AnoGAN [19]	0.966	0.992	0.850	0.887	0.894	0.883	0.947	0.935	0.849	0.924	0.9127
DSVDD [12]	0.980	0.997	0.917	0.919	0.949	0.885	0.983	0.946	**0.939**	0.965	0.9480
OCGAN	**0.998**	**0.999**	0.942	**0.963**	**0.975**	**0.980**	**0.991**	**0.981**	**0.939**	**0.981**	**0.9750**

2 is used, our method has not only registered a better average AUC value, it has reported best AUC for individual classes in 9 out of 10 classes.

CIFAR10 : CIFAR10 is an object recognition dataset that consists of images from 10 classes. Out of the considered datasets, CIFAR10 is the most challenging dataset due to it diverse content and complexity. Specifically, it should be noted that all other datasets are very well aligned, without a background. In comparison, CIFAR10 is not an aligned dataset and it contains objects of the given class across very different settings. As a result, one-class novelty detection results for this dataset are comparatively weaker for all methods. Out of the baseline methods, [12] has done considerably better than other methods. Following their work, we carried out the same pre-processing in our experiments. In addition, we subtracted the class-mean image from all training and testing images. We obtained comparable results to deep-SVDD with the proposed method where we recorded average AUC of 0.6566.

Table 5-III. One-class novelty detection results for CIFAR10 dataset using Protocol 2.

	AIRPLANE	AUTOMOBILE	BIRD	CAT	DEER	DOG	FROG	HORSE	SHIP	TRUCK	MEAN
OCSVM [20]	0.630	0.440	0.649	0.487	0.735	0.500	0.725	0.533	0.649	0.508	0.5856
KDE [13]	0.658	0.520	0.657	0.497	0.727	0.496	**0.758**	0.564	0.680	0.540	0.6097
DAE [15]	0.411	0.478	0.616	0.562	0.728	0.513	0.688	0.497	0.487	0.378	0.5358
VAE [38]	0.700	0.386	**0.679**	0.535	**0.748**	0.523	0.687	0.493	0.696	0.386	0.5833
Pix CNN [78]	**0.788**	0.428	0.617	0.574	0.511	0.571	0.422	0.454	0.715	0.426	0.5506
GAN [19]	0.708	0.458	0.664	0.510	0.722	0.505	0.707	0.471	0.713	0.458	0.5916
AND [23]	0.717	0.494	0.662	0.527	0.736	0.504	0.726	0.560	0.680	0.566	0.6172
AnoGAN [19]	0.671	0.547	0.529	0.545	0.651	0.603	0.585	0.625	0.758	0.665	0.6179
DSVDD [12]	0.617	**0.659**	0.508	0.591	0.609	**0.657**	0.677	**0.673**	0.759	**0.731**	0.6481
OCGAN	0.757	0.531	0.640	**0.620**	0.723	0.620	0.723	0.575	**0.820**	0.554	**0.6566**

Ablation Study

In order to investigate the effectiveness of each additional component of the proposed work, we carried an ablation study using the MNIST dataset. Specifically, we consider four scenarios. In the first scenario we consider only the auto-encoder. In the second and third scenarios, we use auto-encoder with the visual and latent discriminators respectively. In the final scenario, we consider the full proposed model, OCGAN. Mean AUC for each class of MNIST dataset is tabulated in Table 6-V.

We note that the AUC value obtained for the auto-encoder is already high at 0.957. Therefore even slightest of improvement from this point is significant. When a latent discriminator is introduced, performance of the system improves marginally by 0.2%. When a visual discriminator is added on top, the performance improves further by 1%. When informative-negative mining as added, performance is further improved by a 0.4%.

Table 5-IV. Ablation study for OCGAN performed on MNIST.

Without any Discriminators	0.957
With latent Discriminator	0.959
With two Discriminators	0.971
Two Discriminators + Classifier	0.975

Summary

In this chapter, we dived deep into mechanics of reconstruction-error based one-class novelty detection. We showed that a network trained on a single class is capable of representing some out-of-class examples, given that in-class objects are sufficiently diverse. In order to combat this issue we introduce a latent-space-sampling-based network-learning procedure. First we restricted the latent space to be bounded and forced latent projections of in-class population to be distributed evenly in the latent space using a latent discriminator. Then, we sampled from the latent space and ensured using a visual discriminator that any random latent sample generates an image from the same class. Finally, in an attempt to reduce false positives we introduced an informative-negative mining procedure. We showed that our OCGAN model outperforms many recently proposed one-class novelty detection methods on four publicly available datasets. Further, by performing an ablation study we showed that each component of the proposed method is important for the functionality of the system.

Chapter 6

Multiple-class Novelty Detection/ Open-set Recognition

In this chapter, we consider multiple-class novelty detection problem in the absence of out of distribution data[1]. Since there exists annotated data belonging to multiple classes, it is possible to train a deep-classification network in this setting. A naive solution to the stated problem is to threshold the probability of the most probable class produced by the network [79]. CNNs are trained with the objective of maximizing the probability of the correct class over the training data. Therefore, if the training process generalizes well enough, query samples from known classes can be expected to produce high probabilities. However, the open-set recognition literature [9] points out the possibility of novel object samples producing equally high probabilities.

When a discriminative classifier is trained, it learns a set of features that are needed to discriminate between the known classes. In the ideal case, features that are not essential to separate the known classes are discarded during the learning process. We refer to these features as *optimal closed-set features*. However, *optimal closed-set features* are likely insufficient for capturing differences between open-set samples and known-classes [39]–additional features are likely required to separate the known classes and open-set samples. Open-set samples could end up producing high class-activations,

---

[1] In this setting, *open-set recognition* and *multiple-class novelty detection* have been used interchangeably in the literature.

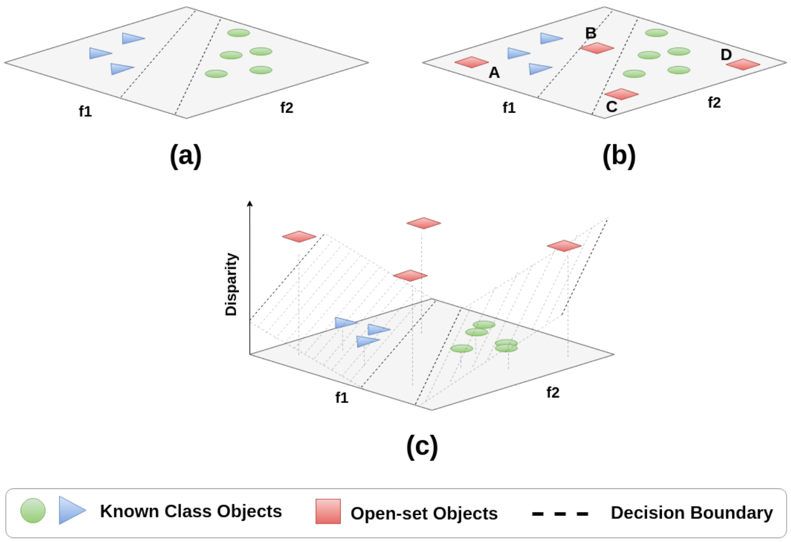


Figure 6-1. (a) A classifier defines a positive half space for each class. (b) An open-set object could project either near a decision boundary (samples B and C) or deep into the positive half space (samples A and D) of a given class. (c) We learn a classifier which takes into account more factors than just class separation.

depending on where in the feature space they are projected.

We investigate two techniques that reduce this effect. First, we extend *optimal closed-set features* so that features have the capacity to describe shapes, structure and semantics of known-class objects. During training, the classifier will consider the overall semantics of images (not just the discriminative aspects) when class decision boundaries are defined. As a result, open-set images will not be positioned in any of the positive half-spaces on the grounds of having different semantics. We obtain such diverse features by incorporating self-supervision in learning.

Second, we model the known-class objects using a generative model. Then, a classifier is learned by considering both the input image and its generative representation. The classifier will take into account the correspondence between the two inputs when the decision boundaries are obtained as shown in Figure 7-2(c). Since the generative model is trained using known-class images, it will not represent open-set samples well. As a result, open-set samples will demonstrate high disparity (Figure 7-2(c)) thereby getting projected out-side the positive half spaces of known classes.

Proposed Method

In this section, we motivate the need for a richer feature representation for effective open-set recognition. Then, we introduce conditioning on generative representation and self-supervision to overcome this challenge. Finally, we describe the proposed training and testing procedure.

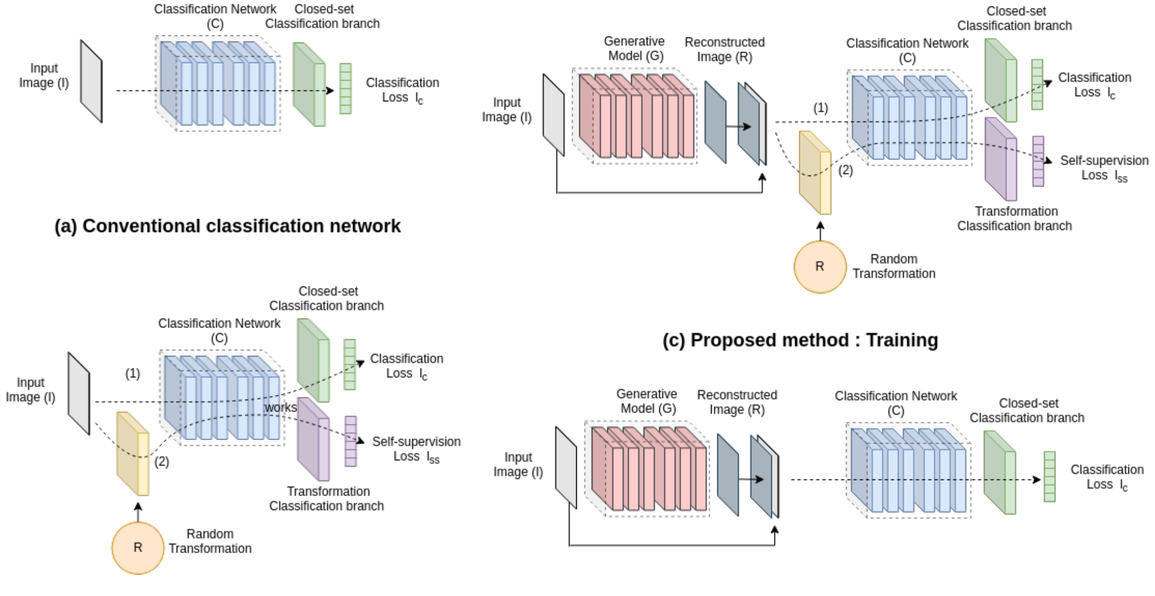
Challenges in Open-set Recognition

An illustration of why open-set recognition is challenging is shown in Figure 7-2. When a classifier is trained, the positive half spaces of each class are identified (these half spaces are described by the vector defined using the final fully connected layer weights corresponding to the class). When a sample appears deeper in the identified positive half space, it will generate a larger class activation. On the other hand, a sample appearing near the half-space boundary will result in a lower class activation. When the network is trained, a feature embedding is learned such that each training sample is encouraged to be pushed deeper in to the positive space corresponding to its ground truth. Therefore, as long as the query samples follow the same distribution as the training samples, known-class samples are expected to produce large activation values.

Consider an open-set image that is projected onto one of the following regions:

1) Intersection of all class boundaries. This will arise when the open-set image does not have any components/regions common with any of the known classes (See points B and C in Figure 7-2)(b). In this case, the class activation scores of all classes will be low. These types of open-set samples may be filtered by thresholding the maximum class activation score.

2) Deep into the positive half space of a class. This situation (such as points A and D in Figure 7-2(b)) arises when the open-set image has a semantically similar component/region to that of a known class (or the network perceives to be so). As a result, the activation of the aforementioned class becomes high. These instances cannot be easily rejected by considering class activations. We specifically focus on the latter case and investigate techniques that can reduce class activation scores of open-set samples.


Input Image (I)
Classification Network (C)
Closed-set Classification branch
Classification Loss l_c
Generative Model (G)
Reconstructed Image (R)
(1)
(2)
Self-supervision Loss l_{ss}
Transformation Classification branch
R
Random Transformation
works

(a) Conventional classification network

(c) Proposed method : Training

(b) Classification network with Self-supervision

(d) Proposed method : Testing

Figure 6-2. Comparison of the network architectures. (a) A CNN network. (b) CNN with self-supervision. (c) Proposed network.

Self-Supervised Learning

When a closed-set classifier is trained, the classifier learns only features that are necessary to differentiate between known classes. However, these features are not always descriptive enough to separate out open-set samples from known classes. By introducing a more descriptive feature, we reduce the activation magnitude of open-set samples. For this purpose, we extend the conventional classification network into a multi-task network where an auxiliary classifier performs self-supervision.

We adopt the self-supervision framework proposed in [80]. In [80], a geometric transformation is applied to an input at random from a finite set of transformations, and the self-supervision branch of the network is used to predict which transformation was applied. In order to determine the transformation that was applied, the network needs to learn structural properties of image content such as shape and orientation. As a result, when a self-supervision branch is added on top of the classification task, the intermediate features becomes more descriptive.

Figure 6-2(a) and (b) illustrate network architectures of a conventional classification network and a classification network extended to perform self-supervision respectively. In the former case, each training instance is passed through the classification network (C) to produce a classification loss l_c. In the latter case, the classification network (C) has two output branches. Each forward pass consists of two steps. In the first step, a classification loss l_c is produced by passing the input through the open-set classification branch. During the second step, the input image is subjected to a random transformation. The transformed image is passed through the transformation classification branch to arrive at self-supervision loss l_{ss}. When evaluating the self-supervision loss, the transformation applied to the input is considered to be ground truth. The network is trained by considering a composite loss of the form $\alpha_1 l_c + \alpha_2 l_{ss}$. In our experiments, we chose $\alpha_1 = 0.8, \alpha_2 = 0.2$ with the aim of giving more importance to the primary classification task[2]. For our experiments we used 14 transformations where each transformation was formed by randomly flipping the image (horizontally and vertically) and by rotating image by multiples of 90 degrees.

In section 6, we demonstrate the effectiveness of introducing self-supervision through an ablation study.

Augmenting with Generative Representation

As the second contribution of our work, we augment the input with its representation obtained through a generative framework. Let us first consider a generative model

---

[2] Please refer to the supplementary material for a sensitivity analysis of these parameters.

trained on the images of known classes. For example, the generative model can be a deep auto-encoder network. Ideally, the generative model will be able to represent and reproduce samples of known classes. On the other hand, since the generative model has not seen samples from open-set classes, it will not be able to represent (re-produce) such samples equally well. If this is the case, there will be high correspondence between input images and reconstructed images generated by the generative model for known class samples. Correspondence will be low for open-set samples.

In Figure 7-2(c), we illustrate the implication of augmenting a generative reconstruction to the open-set problem. In this idealistic case, we have denoted the disparity between the original image and the reconstructed image as an additional axis. Here, the disparities for known samples are smaller compared to open-set samples. In this scenario, the classifier will learn two new positive half planes defined by hyper-planes similar to that of shown in Figure 7-2(c). If disparity is considerably high, it will force opens-set samples to be outside the positive half space of all the classes.

Based on this intuition, we carry out the training process in two steps. First, we train a generative network (G) using training samples. Then, given an input x, we train the classification network (C) by considering the augmented input $[x, G(x)]$ as shown in Figure 6-2(c).

Training and Testing Procedure

Architecture. We use the network architecture proposed in [43]. The encoder network used for the autoencoder consists of 10 convolutional 3×3 layers, where each layer is followed by a batch-normalization and leaky ReLu(0.2) operation. The decoder network has a similar structure to that of the encoder and is constructed with transpose-convolution layers instead. The classifier network consists of 9 layers of 3×3 convolution filters followed by batch-normalization and leaky ReLu(0.2) operations. It is terminated using a fully-connected layer. The only difference in our classifier from

Generative Model	**Classifier Model**	**Open-set Performance**					
		1	**2**	**3**	**4**	**5**	**Avg**
Vanilla AE	Vanilla CNN	78.0	76.7	84.9	84.9	79.4	80.8
Conditioned-AE	Vanilla CNN	79.1	77.3	85.7	87.4	80.3	82.0
Conditioned-AE	WRN28-10	77.5	81.7	86.2	87.5	82.6	83.1
WGAN	WRN28-10	81.7	79.2	85.5	87.2	84.3	83.6

Table 6-I. Impact of using different architectures on open-set recognition on the CIFAR10 dataset. We observe that using more sophisticated generative models and classifiers both improve open-set performance.

[43] is that our network accepts a 6-channel image as the input.

In order to investigate the impact that different architectures have on open-set rejection performance, we vary the classifier and generative model and study the impact they have on open-set recognition. In Table 6-I, we tabulate open-set recognition performance in terms of AUC-ROC under different architectures across five different known-openset splits for the CIFAR10 dataset. Here, vanilla AE and vanilla CNN refers to the network architectures used in [43]. Conditioned AE [81] is a modified version of Vanilla AE, where a fully connected layer classifier is connected to the latent space. This version of the AE produces better known-openset separation in reconstructed image space due to this additional constraint. WRN28-10 and WGAN refers to standard wide-ResNet(depth 28 and width 10) [82] and Wasserstein GAN [83] respectively. According to Table 6-I, we observe that using a more sophisticated network, both as a generative model and a classification model have contributed towards improving average open-set recognition performance.

Training. We trained all networks for 1000 iterations using the Adam optimizer with a batch size 64, learning rate of 0.001 and parameters $(0.5, 0.999)$. The training process is outlined in Figure 6-2(c) and Algorithm 2. As descried in Algorithm 2, each training sample is first passed through the classification branch of the network to obtain the classification loss l_c. Then, a transformation is randomly selected from the set of available transformations. If the chosen transformation index is r, the transformed image is passed through the self-supervision branch to produce a self-supervision loss

```
Input  : Training sample $x$,label $y$, Transformation Set $T$, Models: $G, C$ ,
         Weights $\alpha_1$,$\alpha_2$
Output : Models: $G, C$
Classification Step.
$\hat{x} \longleftarrow G(x)$
$z = [x, \hat{x}]$;
$l_c$ = CrossEntropy$(C(z), y)$
Self-supervision Step.
Pick transformation randomly.
$r$ = rand$(\Omega(T))$
$t = T[r]$;
$z = [t(x), t(\hat{x})]$;
$l_{ss}$ = CrossEntropy$(C(z), r)$
$l_t = \alpha_1 l_c + \alpha_2 l_{ss}$
Backpropagate to change $G$ and $C$.
```

Algorithm 2: Training Algorithm

which is calculated using cross-entropy by considering r as the ground-truth label. The composite loss t_l is backpropagated to find gradients associated with each network weight. Finally, networks C and G are updated according to the network updating algorithm.

Testing. During inference, the self-supervision branch of the network is disregarded as shown in Figure 6-2(d). Given a query image x, first the augmented representation $[x, G(x)]$ is obtained. Then, the augmented input is passed through the classifier network to obtain class activations $a = C([x, G(x)])$. If the maximum activation $max(a)$ is below a predetermined threshold γ, it is declared that the input is an open-set instance. In practice, threshold γ is determined such that a minimum true positive rate is guaranteed on a validation set. In our experiments we picked γ such that true positive ratio is at least 0.9.

Experimental Results

We evaluate the performance of the proposed method on standard datasets used for open-set recognition and compare with state-of-the art methods. First, we report performance on open-set recognition and out-of-distribution recognition tasks respectively. Then we consider a case study on the CIFAR10 dataset to analyze performance of the proposed method qualitatively. We conclude the latter section with an ablation study.

Open-set Recognition

Recent deep learning based open-set recognition methods followed the protocol in [43] and used the numbers reported in [43] as a baseline for comparison. In [43], an open-set recognition scenario is simulated on a multi-class classification dataset by randomly selecting n classes as known. The remaining classes are considered to be open-set classes. This protocol is used to simulate five trials of open-set recognition and performance is measured using the average area under the curve of ROC (AUC-ROC) curve.

Performance across different splits varies significantly (in our experiments AUC for CIFAR10 varied between 77% to 87% across different splits). There are many possible known-openset combinations one could consider when the above protocol is followed($\binom{10}{6}$ for CIFAR10, SVHN and $\binom{200}{20}$ for TinyImageNet). Open-set performance is highly correlated with the classifier performance. A better classifier is able to reject open-set samples more effectively (for example in [45], open-set performance improves when a DenseNet backbone is used as compared to a vanilla CNN). Therefore, for a fair comparison, we argue that all methods should use identical splits and the same network backbone.

In this spirit, we use the same autoencoder and classifier architectures as [43]. Further, we test on the same known-openset splits as [43][3]. Note that [44] used different known-openset splits in their evaluation. We used the code released by the authors of [44][4] to evaluate open-set performance on the same splits and we report

---

[3] Exact splits used by [43] can be found at github.com/lwneal/counterfactual-open-set.

[4] Code is found at github.com/otkupjnoz/c2ae. We validated results obtained for considered class splits with authors of [44].

these results in our paper.

We carried out tests on the following datasets using the protocol described in [43]:

CIFAR10 and SVHN. Both CIFAR10 [65] and SVHN [84] are 10-class classification datasets. CIFAR10 contains data from four vehicle classes and six animal classes. SVHN is a dataset of photographed numbers. In our tests we considered splits from [43] where six classes are chosen to be known. Remaining classes are considered to be open-set.

CIFAR+10. CIFAR+10 training set consists vehicle classes of CIFAR10 dataset as known-classes. Vehicle classes from CIFAR10 and 10 vehicle classes samples from CIFAR100 [85] is considered to be open-set classes.

CIFAR+50. Same training setting as CIFAR10+. The vehicle classes from CIFAR10 and 50 vehicle classes samples from CIFAR100 are considered to be open-set classes.

TinyImageNet. TinyImageNet is a sub-set of 200 classes taken from the ImageNet dataset [63]. 20 classes are considered to be known and remaining 180 classes are considered to be open-set. Known-openset splits are chosen to be the same as in [43].

In Table 6-II, we tabulate open-set detection performance of known-classes for the proposed method with baseline methods. For each experiment, we indicated the *openness*[9], defined by $1 - \sqrt{\frac{K}{M}}$, where K and M denote the number of known classes and total number of classes, respectively. The performance of the baseline methods is obtained from [45] and [43]. According to Table 6-II, the proposed method has a significant improvement for the CIFAR10 dataset with an increase in performance of over 10%. A similar improvement is seen for the CIFAR+10 and CIFAR+50 test cases. Since CIFAR+50 dataset has more openness due to more open-set classes, it has produced slightly lower performance compared to CIFAR+10. For the SVHN dataset, the performance improvement is about 2%. For TinyImageNet, our performance is on par with other open-set methods where the proposed method performs marginally better. Table 6-III lists the closed set classification accuracy

	CIFAR10 13.39%	CIFAR+10 33.33%	CIFAR+50 62.86%	SVHN 13.39%	TinyImageNet 57.35%
SoftMax	67.7± 3.8	81.6± *N.R.*	80.5± *N.R.*	88.6± 1.4	57.7± *N.R.*
OpenMax (CVPR16) [**Bendale_2016_CVPR**]	69.5±4.4	81.7± *N.R.*	79.6± *N.R.*	89.4±1.3	57.6± *N.R.*
G-OpenMax (BMVC17) [42]	67.5±4.4	82.7±N.R.	81.9± *N.R.*	89.6±1.7	58.0± *N.R.*
OSRCI (ECCV18) [43]	*69.9±3.8*	*83.8± N.R.*	*82.7± N.R.*	*91.0±1.0*	58.6± *N.R.*
C2AE (CVPR19) [44]	68.2 ± 4.1	82.3 ± 0.3	81.3 ± 0.3	89.3 ± 1.6	58.1 ± 1.9
CROSR(CVPR19) [45]	*N.R.*	*N.R.*	*N.R.*	89.9±1.8	*58.9±N.R.*
Ours (Plain CNN)	**80.7±3.9**	**92.8±0.2**	**92.6±0.0**	**93.5±1.8**	**60.8±1.7**
Ours (WRN-28-10)	83.1±3.9	91.5±0.2	91.3±0.2	95.5±1.8	64.7±1.2

Table 6-II. Open-set detection performance in terms of AUC-ROC curve. N.R. is used when the original work did not report a particular figure.

	CIFAR10	CIFAR+10	CIFAR+50	SVHN	TinyImageNet
Ours (Plain CNN)	92.8±1.7	94.4±0.0	94.4±0.0	96.6±0.4	49.2±2.9
Ours (WRN-28-10)	95.09±1.3	97.4±0.2	97.4±0.2	97.29±1.3	55.9±2.8

Table 6-III. Closed-set accuracy for the proposed method.

for each dataset. In both Tables 6-II and 6-III, we reported the performance of our method when WideResNet28-10 [82] classifier is used. It can be observed that using WideResNet, which is a better classifier, open-set recognition performance increases in majority of time. This result suggests that better performance can be obtained by using more sophisticated classifiers.

Out-of-distributional Detection

We evaluate the performance of the proposed method in Out-of-distributional detection (OOD) [79] on CIFAR10 dataset. Out-of-distributional detection is a special case of open-set detection. Here, it is assumed that the open-set samples follow a different distribution than the known-set distribution. Following the protocol outlined in [45], we considered all classes in CIFAR10 as known-classes and trained a 13-layer VGG model as specified in [45]. The output channels of each 3×3 convolutional block number were 64, 128, and 256, and they consist of two, two, and four convolutional layers with the same configuration. Then, we consider test images from ImageNet and LSUN dataset [86] as out-of-distributional images when each are cropped and resized respectively [87].

Table 6-IV shows the out-of-distributional performance in terms of macro-averaged

Training Method	Detector	ImageNet-Crop	ImageNet-Resize	LSUN-Crop	LSUN-Resize
Cross-entropy	SoftMax [79]	63.9	65.3	64.2	64.7
	OpenMax [**Bendale_2016_CVPR**]	66.0	68.4	65.7	66.8
Counterfactual	SoftMax [43]	63.6	63.5	65.0	64.8
LadderNet	SoftMax [79]	64.0	64.6	64.4	64.7
	OpenMax [**Bendale_2016_CVPR**]	65.3	67.0	65.2	65.9
	CROSR [45]	62.1	63.1	62.9	63.0
DHRNet	SoftMax [79]	64.5	64.9	65.0	64.9
	OpenMax [**Bendale_2016_CVPR**]	65.5	67.5	65.6	66.4
	CROSR [45]	72.1	73.5	72.0	74.9
Ours	Activations	*75.7*	**79.2**	*75.1*	**80.5**
	SoftMax	**82.1**	*77.7*	**84.3**	*78.4*

Table 6-IV. Performance of out-of-distributional object detection for CIFAR10 dataset with VGG13 network. Performance is measured using macro-F1 measure.

F1 score. For the proposed method, following other OOD works [87], every sample producing a score lower than a 10%th percentile of matched scores were identified as open-set. It should be noted that it is customary to detect OOD samples based on SoftMax scores [79]. Therefore in Table 6-IV we reported F1 scores for the proposed method both when SoftMax scores and class activations are considered for decision making. All other numbers except ours are taken from [45]. According to Table 6-IV, the proposed method out-performs baseline methods in all test cases. It should be noted that SoftMax scores yielded better OOD detection compared to class activation scores whenever images are cropped instead of resized. This is not surprising as an image crop contains little structure. As a result, image crops are more likely to produce balanced probabilities thereby making open-set detection based on SoftMax probabilities more effective.

Case Study and Ablation Study

We conducted a case-study on CIFAR10 dataset where all animal classes (bird, cat, deer, dog, frog and horse) were considered to be known. Vehicle classes (airplane, car, ship and truck) were considered to be open-set. We compare the performance of a conventional CNN network (Figure 6-2(a)) with the proposed method (Figure 6-2(c)). The conventional CNN produced a AUC of 84.35% where as the proposed method produced an AUC of 91.24%.

Figure 6-3 visualizes the score histograms generated for open-set samples and

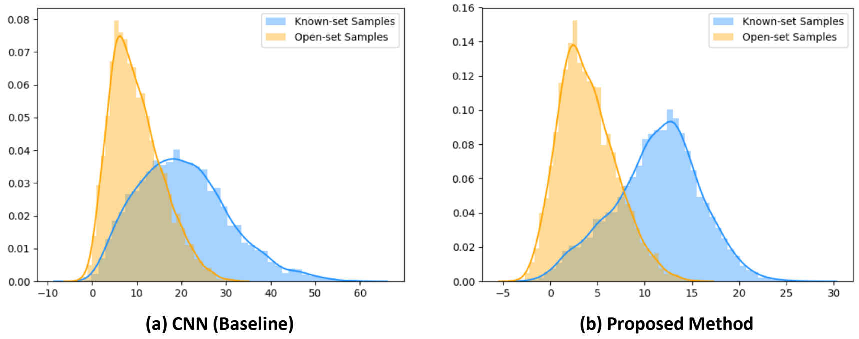
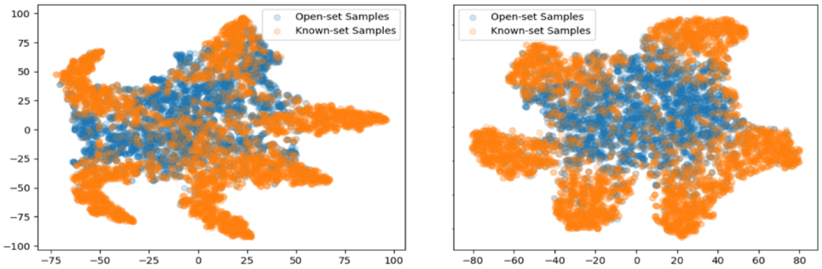


Figure 6-3. Score histograms for open-set and known-set samples.

known-class samples for both methods. As evident from Figure 6-3, the proposed method has better score separation between open-set and known-set samples. This is why a larger AUC value has been obtained from the ROC curve for the proposed method.

To understand why a better score separation was obtained, we visualized the final feature space for both baseline CNN and the proposed method using tSNE [88] in Figure 6-4. In both cases, six clusters can be observed in the tSNE visualization plane in Figure 6-4; these clusters correspond to each class. However, there is a considerable over-lap between known-set samples and open-set samples in the baseline CNN (Figure 6-4(a)). On the other hand, under the proposed scheme (Figure 6-4(b)) overlap between known and open-set samples are less. Further we note that known clusters appearing under the proposed method is more compact compared to the baseline case. This is because proposed method models the whole data distribution (as a result of self-supervision and generative feature augmentation) as opposed to modeling just the boundary as usually done in conventional CNNs. The proposed method has a lower overlap between known and open-set samples in the feature space. Therefore, it produced better separation between known and open-set distributions as shown in Figure 6-3.

In Figure 6-5, we show eight open-set images that had produced the largest



(a) CNN (Baseline) **(b) Proposed Method**

Figure 6-4. tSNE visualization of the feature space for (a)Conventional CNN and for the (b) proposed method.

activations in the baseline CNN. It should be noted that although these images have generated high score activations, none of them have a close resemblance to any of the known-set of classes. In the same figure, we illustrate class activation scores obtained by the baseline CNN (middle column) and the proposed method (right column). Since the range of activation scores is different under the two methods, as Figure 6-3 shows, for a fair comparison we have normalized these scores using z-score normalization by considering all open-set scores under each scheme.

According to Figure 6-3 (Middle), the baseline CNN has produced a score around 2 for all samples. On the other hand, under the proposed scheme, the same images have generated lower scores. Except for the third and fourth images, activations produced by all other images have been reduced by at least by a factor of half. This example illustrates that the proposed method has even lowered activations for hard open-set samples.

Finally, it is worth noting the contribution each component of our proposal has towards the final outcome of the he algorithm. In order to assess this, we carried out an ablation study on CIFAR10 by considering animal classes as known-set classes. We considered following cases.

Baseline. The classifier network operating on only the input images as shown in

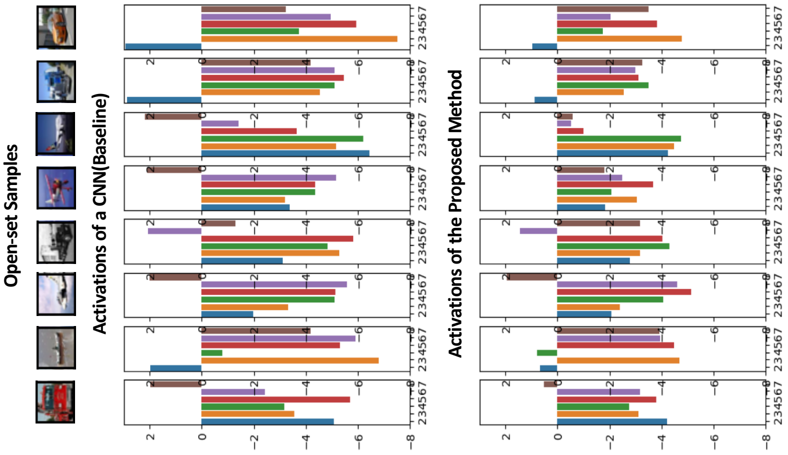


Figure 6-5. Top Row: Visualization of open-set samples. Middle Row: z-score normalized activations produced by the baseline CNN. Bottom Row: z-score normalized activations generated by proposed method.

Figure 6-2(a).

Self-supervision. Classification network extended to perform self-supervision as shown in Figure 6-2(b).

Augmented Classifier. Generative feature is used to augment the input image space. A classifier is trained on the augmented input. No self-supervision is used.

Proposed method. Classifier is learned on augmented image space with self-supervision (Figure 6-2(c)).

In Table 6-V we report closed-set classification accuracy along with open-set rejection performance in AUC-ROC. According to Table 6-V, the baseline produced a AUC-ROC value of 84.0%. The introduction of self-supervision and augmented features both independently improved open-set performance by 4%, where improvement induced by augmented features is marginally better than self-supervision. Finally, when both techniques are combined (the proposed method), performance further improves by 2.7% to arrive at 91.2%. This study demonstrates that each component of the proposal is contributing towards the final performance boost that is observed.

	Classification Accuracy	**Open-set Rejection(AUC)**
Baseline	89.7	84.4
Self-supervision	92.4	88.8
Augmented Classifier	91.5	88.4
Proposed Method	92.6	91.2

Table 6-V. Tabulation of classification performance in terms of accuracy and open-set rejection performance(AUC) for the ablation study.

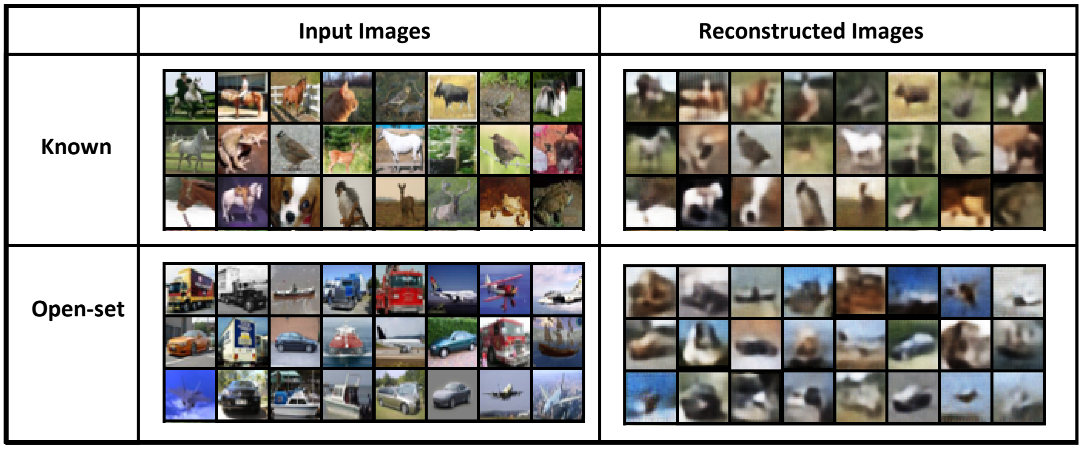


Figure 6-6. Reconstructed images produced by the auto encoder trained on known-set images.

In Figure 6-6 we visualize reconstructions (of randomly chosen samples) obtained through the generative model. According to Figure 6-6, all reconstructed images take the form of a blurry version of the input images. However, we note that known-set samples carry more details compared to open-set classes. For an example, it is hard to predict the class label of open-set classes by merely looking at the reconstructed image. However, the amount of information preserved in the reconstructed image is not a very good indicator to detect open-set images (AUC is merely 66.7% when it is used as an indicator). Nevertheless, it provides information that can be leveraged to make a better informed decision.

Summary

We explore the detection of open-set samples more effectively by learning richer feature representations than are usually needed for closed-set classification. We used self-supervision and augmented the input image with a representation obtained from a generative model to enhance network's ability to reject open-set samples. These improvements forced the classifier to look beyond what is required to perform closed-set classification when producing decision regions. We evaluated the proposed method in open-set detection and out-of-distributional image detection experiments where we produced state-of-the-art results.

We carried out a study investigating the importance of each component of the proposed method. Further, we demonstrated qualitatively how proposed method results in better separation in feature space thereby producing lower activations for open-set samples. Finally, we experimented with different choices of generative models and classifiers, where we concluded that using more sophisticated models in both cases would benefit open-set detection performance.

Chapter 7

Data Efficient Novelty Detection with Low Latency

In this chapter, we study a practical application of novelty detection in mobile Active Authentication (AA). Mobile device security has become one of the major concerns in modern day life due to sensitive information they contain. Active Authentication is a device authentication method that essentially make use of the physiological and behavioral biometrics using built-in sensors and accessories such as gyroscope, touchscreen, accelerometer, orientation sensor, and pressure sensor to continuously monitor the user identity [5]. In the decision making process, AA boils down to the problem of novelty detection – where the enrolled users of the device become *known classes* of the device. The goal of AA is to detect when an intruder (a person outside the enrolled set of users) starts using the device. By definition, an intruder in this context becomes a novel class instance. Therefore, AA can be solved using a novelty detector trained on the enrolled users.

When designing a practical novelty detector for AA, one has to consider various factors such as security and usability. It is well known that a balance needs to be made between security and usability of a biometrics-based AA system [89–91]. The design of usable yet secure AA systems raises crucial questions concerning how to solve conflicts between mobile security and usability. In order to balance usability and security of an AA scheme, we must address the following fundamental challenges.

1. Accuracy : How accurately does a mobile AA system detect an attacker or an intruder? Due to limitations of representation and classification models on mobile devices, behavioral and physiological biometrics-based methods do not provide good accuracy in practice. The AA system will be of little use in terms of security if it produces a high degree of false positives. On the other hand, a higher false negative rate would severely degrade the usability of the technology. Many recent approaches in the literature have attempted to address this factor by proposing better features and classifiers.

2. Latency : How long does it take to detect an attacker? If an AA system takes too long (e.g. 1-3 minutes) to detect an intrusion, it would grant an intruder plenty of time to extract sensitive information prior to the lock down. Hence, unless intruder detection is sufficiently fast, the AA system would hold a little value in practice no matter how high its detection accuracy is.

Consider a series of observations captured from a front-facing camera of an Android device shown in Figure 7-1. Frames (A-I) belong to the genuine user of the device. From frame J onwards an attacker starts to operate the device. In this scenario, frame J signifies a change point (i.e. an intrusion). The AA system should be able to detect intrusions with a minimal delay while maintaining a low rate of false detections. For instance, note the changes in genuine user's images in frames (D-F) due to camera orientation and facial expressions. While having a fast response, an AA system ideally should not falsely interpret these variations as intrusions.

3. Efficiency : How much resource does the system use? By definition, mobile AA systems are continuous processes that run as background applications. If they consume considerable amount of resources, memory and processing power, it could slow down other applications and cause the battery to drain quickly. Despite the improvements in mobile memory and processors, battery capacity remains to be a

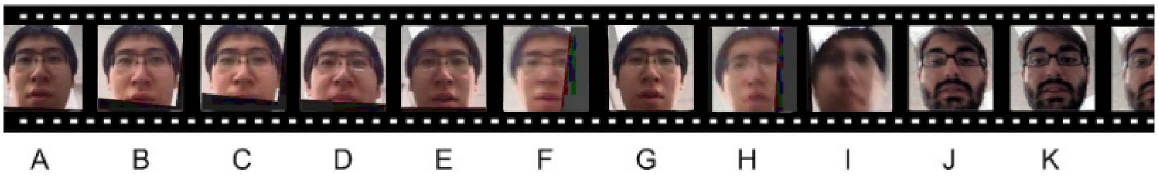


Figure 7-1. The problem of quick intrusion detection in face-based AA systems. (A-I) show the genuine user with varying facial expressions. An intrusion occurs starting from (J).

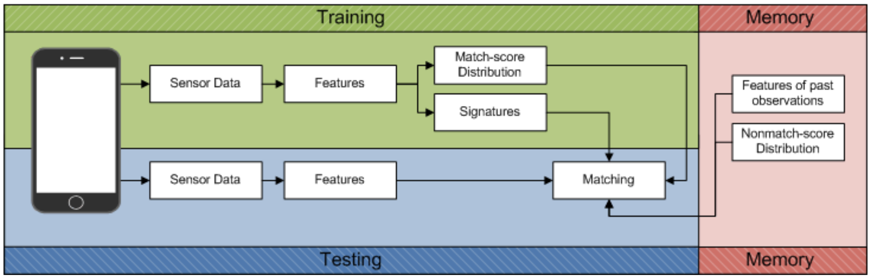


Figure 7-2. An overview of the proposed QCD-based AA method.

constraint due to limitations in heat transfer and space. Therefore, it can be expected to be the bottleneck in terms of efficiency in years to come. If an AA application causes battery to drain too quickly, then it is unrealistic to expect the users to use AA technology as they would typically opt out from using such applications [92]. Therefore, efficiency has a huge impact over the usability of AA as a technology. Recently, [93] studied the efficiency of a mobile AA system based on face biometric. Experiments were conducted on a Google Nexus 5 device with 2GB of RAM and a quad core 2.2GHz CPU. It was shown that the normal usage of the device consumes about 520 mW of power and the facial attribute-based AA framework running at 4 frames per second consumes about 160.8mW additional power. It is needless to say that nearly 30% increase in power consumption would take a toll on battery duration. A trivial solution for this problem would be to decrease the sampling rate of data acquisition. However, effects of such a measure on the detection performance have not been studied in the literature.

Many existing AA systems attempt to improve the accuracy of the system by proposing sophisticated features and classifiers. However, how fast an AA system could detect an intruder has not been widely studied in the literature. Yet, it remains to be an important feature of an AA system. In this paper, we address the problem of quickly detecting intrusions with lower false detection rates in mobile AA systems. We propose Quickest Change Detection (QCD), which is a well-studied problem in statistical signal processing and information theory, for the purpose of intrusion detection in mobile AA systems. Figure 7-2 gives an overview of the proposed method. As opposed to a conventional AA system, the proposed system utilizes all past observations along with distributions of match and non-match data of the genuine user to arrive at a decision. The proposed method does not require a specific feature nor a specific classifier; therefor it can be built upon any existing AA system to enhance its performance.

Table 7-I. Notations used in this paper.

Notation	Definition
x_i	Match score obtained at the i^{th} time instance
f_0	Density of matched scores
f_1	Density of non-matched
$E[.]$	Expectation operator
$P[.]$	Probability function
$(x)^+$	Positive portion of x
T	Time at an intrusion occurs
π_n	Probability of intrusion occurring at time n
ρ	Probability of an intrusion occurring
C_α	Set of possible solutions for threshold α
p_n	Probability of change has occurred at time n
$L(.)$	Likelihood ratio
M_i	Indicator of whether observation i is recorded

Table 7-II. List of abbreviations.

Abbreviation	Meaning
AA	Active authentication
ADD	Average detection delay
ANO	Average number of observations
APO	Average percentage of observations
BQCD	Bayesian Quickest change detection
CDC	Change duty cycle
CPU	Central processing unit
E-BQCD	Efficient Bayesian quickest change detection
E-MQCD	Efficient minimax quickest change detection
FAR	False acceptance rate
LBP	Local binary pattern
MQCD	Minimax Quickest change detection
PFD	Probability of false detections
PIN	Personal identification number
QCD	Quickest change detection
RAM	Random access memory
WADD	Worst average detection delay

Intruder Detection in AA

A typical AA system consists of several stages as illustrated in Figure 7-3. Initially, sensor data of the genuine user is obtained through an enrollment phase and a set of features are extracted from the enrolled data. Face images, swipe gesture coordinates, gyroscope/ accelerometer readings and microphone amplitudes are popular choices of data for this purpose. These set of features serve as the gallery at the matching stage.

Upon the initial login of the user, the device continuously collects the same set of data as before during the normal operation of the device. This stage is the Data Acquisition phase shown in Figure 7-3. Features generated with the collected data are compared against the gallery using a Biometric System using a suitable authentication algorithm. At the end of the comparison phase, a match score x_i is obtained. At the n^{th} time instance based on previously observed matched scores $x_1, x_2, ..., x_n$, a decision is made as to whether an intrusion has occurred or not. If an intrusion has occurred, the phone is locked and the user is prompted to verify his/her identity by the means of a primary verification method. This typically takes the form of a password or a primary biometric such as fingerprint. Otherwise, the user is allowed to continue with the device until the next sensor observation.

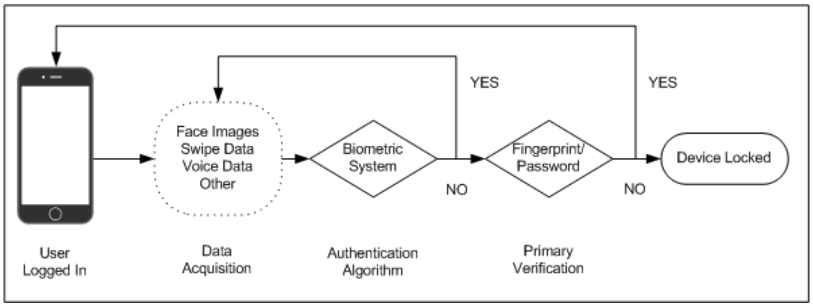


Figure 7-3. An overview of a typical AA system.

The score distribution obtained as explained for the genuine user is henceforth referred to as the match score distribution (f_0). Similarly, score distribution of non-genuine users (intruders or attackers in this context) is referred to as the non-match distribution (f_1). Hence, when an intrusion occurs, the distribution of observations changes from being match to non-match. Therefore, an intrusion point is treated as a change point. With this background, we use the words pre-change distribution and match distribution interchangeably. Similarly, post-change distribution and non-match distribution are used interchangeably. If the match distribution has considerable overlap with the non-match distribution, then the detection results tend to be poor. This is typically the case in mobile devices where sensor data acquisition appears in an unconstrained setup. For example, in the case of face-based AA, face images captured by the front-facing camera contain profile faces, tilted faces as well as partial faces. Therefore, the resulting match score distribution tends to be broad. On the other hand, usage of more sophisticated tools that provides better separation between the two distributions are not preferred for mobile applications due to hardware limitations of the device. As a result, match and non-match distributions tend to overlap considerably. In this context, a more scientific approach backed by a theoretical reasoning is essential to perform the detection of the change. To this end, we propose the use of statistical QCD to detect intrusions in the mobile AA systems.

In the following subsections, we identify two essential characteristics such an AA system needs to possess in order for it to be useful in practice.

Average Detection Delay (ADD)

The primary goal of an AA system is to promptly detect intrusion when the intruder attempts to access the device. Therefore, detection delay of intruder attempts is an important characteristic of a mobile AA system. If the system requires large number of sensor samples to identify an intrusion, there is a possibility that information theft has already occurred by the time intrusion was detected. Hence, from the point view of security [94], it is more desirable to have an AA system with a low intrusion detection delay.

Probability of False Detections (PFD)

On the other hand, if an AA system generates large number of false intruder detections, it would reduce the usability [94] of the user. For example, consider the system shown in Figure 7-3. The AA system prompts the user to enter a password every time AA fails. If the AA system consistently generates false intruder detection alarms, the user will be prompted to enter the password regularly - thereby greatly degrading consumer experience (usability).

As a consequence, Average Detection Delay (ADD) and Probability of False Detections (PFD) play a vital role in any AA system. If T is the real change point, mathematically ADD and PFD at time τ are defined as follows

$$\begin{aligned} ADD(\tau) &= E[(\tau - T)^+] \\ PFD(\tau) &= P[\tau < T], \end{aligned} \tag{7.1}$$

where $E[.]$ and $P[.]$ are the expectation and probability with respect to τ, respectively and $[(x)^+]$ denotes the positive part of x.

From these definitions, one can see that there is an inverse correlation between these two quantities. Generally, obtaining more sensor samples enhances the chance of making a more accurate decision on whether an intrusion has occurred or not. However, this can only be done at the cost of having a relatively larger intrusion detection delay. Therefore, there is always a trade-off between intrusion detection delay and false intruder detection rate. Since, the relationship between ADD and PFD characterizes the performance of an AA system, we propose using the ADD-PFD graph as a tool to compare the performance of different AA systems. Shown in Figure 7-4 are a set of ADD-PFD plots drawn for practical non-sequential AA systems. As expected, in order to obtain very accurate detections (corresponding to a lower PFD), more samples are required to be processed. Moreover, according to Figure 7-4, making a decision based on fewer samples are prone to more false intruder detections.

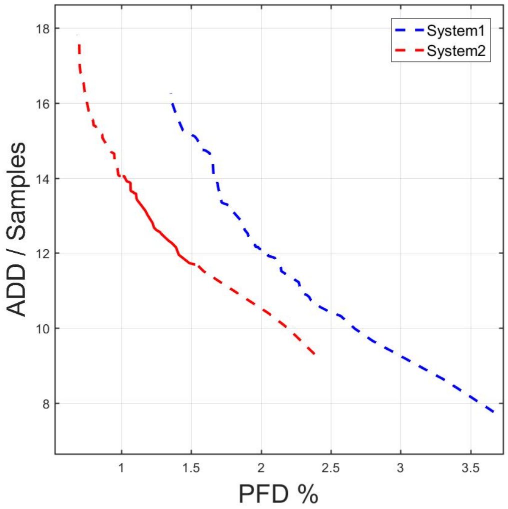


Figure 7-4. Sample PFD-ADD curves of two AA systems.

Adhering to security and usability principles [94], the objective of an AA system is to be able to detect intrusions while ensuring probability of false intruder detection is very low. Therefore, the AA system should operate in a region where both ADD and PFD are comparatively low. For example, for the system represented by the red curve in Figure 7-4, a practical choice would be to operate in the region denoted by the solid line.

Based on this rationale, a better AA system should have a PFD-ADD curve operating below other comparable systems. For example, considering the two operating curves shown in Figure 7-4, system corresponding to the red colored line has a better performance since its operating curve lies at a lower space compared to the other system.

Quickest Change Detection

Quickest Change Detection is a branch of statistical signal processing that thrives to detect the change point of statistical properties of a random process [95], [96], [97]. The objective of QCD is to produce algorithms that detect the change with a minimal delay (ADD) while adhering to false alarm rate constraints (PFD). Consider a collection of obtained match scores, $x_1, x_2, \cdots, x_n$, from the AA system shown in Figure 7-2. Assuming that individual scores are mutually independent, QCD theory can be used to determine whether a change has occurred before time n or not. In the following subsections we present two main formulations of QCD.

Bayesian QCD (BQCD)

In the Bayesian setting [95], it is assumed that the time τ when the change occurs is distributed according to a geometric distribution, Geometric(ρ). Here, the value of ρ is the probability of a change occurring (an intrusion in this context). Conditioned on the change point τ, observations obtained before and after the change follows two distinct distributions, f_0 and f_1. At each time n, based on $\pi_i = P\{\tau = i\}$ for all $i < n$, a decision is made as to whether a change has occurred or not. Based on this formulation, ADD and PFD can be redefined as

$$ADD(\tau) = E[(\tau - T)^+] = \sum_{n=0}^{\infty} \pi_n E_n[(\tau - T)^+] \tag{7.2}$$

$$PFD(\tau) = P[\tau < T] = \sum_{n=0}^{\infty} \pi_n P_n[\tau < T], \tag{7.3}$$

where, for a Geometric(ρ) distribution,

$$\pi_n = P\{\tau = n\} = (1-\rho)^{n-1}\rho$$

for $0 < \rho < 1$ and $n > 0$. Then the Bayesian QCD becomes an optimization problem where the requirement is to minimize ADD subjected to a constraint on PFD. If the class of stopping times adhering to the constrain α on PFD is defined as

$$C_\alpha = \{\tau : PFD(\tau) < \alpha\},$$

then the QCD problem takes the form of Shiryaev's formulation [98], [95]. Objective of the Bayesian QCD formulated by Shiryaev is to obtain a stopping time $\tau \in C_\alpha$ to minimize ADD(τ) for a given α. If p_n is the posterior probability that a change has occurred at time n given observations up to time n

$$p_n = P[T \leqslant n | X^n],$$

where $X^n = (x_1, x_2, ..., x_n)$, then using the Bayes rule, it was shown in [95] that p_n follows a recursive formula as follows

$$p_{n+1} = \Phi(x_{n+1}, p_n),$$

where

$$\Phi(x_{n+1}, p_n) = \frac{\tilde{p_n} L(x_{n+1})}{\tilde{p_n} L(x_{n+1}) + (1 - \tilde{p_n})}.$$

Here, $\tilde{p_n} = p_n + (1 - p_n)\rho$ and

$$L(x_{n+1}) = \frac{f_1(x_{n+1})}{f_0(x_{n+1})}$$

is the likelihood ratio with $p_0 = 0$.

From Theorem 3.1 in [97], this recursive formula provides an optimal solution for the problem in hand with a stopping time of

$$\tau_s = \inf\{n \geqslant 1 : p_n \geqslant A\alpha\}$$

if $A \in (0, 1)$ can be chosen such that $PFD(\tau_s) = \alpha$. This method is known as the Shiryaev test and its proof can be found in [98], [97].

MiniMax QCD (MQCD)

In most of the practical AA systems, probability of intrusion is not known in advance. Therefore, it is important to study QCD in a non-Bayesian setting. MiniMax QCD formulation treats the change point τ as an unknown deterministic quantity [96], [97]. However, as earlier, it is assumed that pre-change distribution, f_0, and post-change distribution, f_1, are known. Due to the absence of prior knowledge on the change point, a reasonable measure of PFD is the reciprocal of mean time to a false detection as follows

$$PFD(\tau) = \frac{1}{E_\infty[\tau]}.$$

Based on this definition of PFD, Lorden proposed a minimax formulation for QCD [99], [96]. Consider the set of stopping times D_α for a given constraint α such that

$$D_\alpha = \{\tau : PFD(\tau) \leqslant \alpha\}.$$

Adhering to this constraint, Lorden's formulation optimizes a cost function to solve the minimax QCD problem. In particular, the cost function is the supremum of the average delay conditioned on the worst possible realizations as follows

$$WADD(\tau) = \sup_{n \geqslant 1} \operatorname{ess\,sup} E_n[(\tau - n)^+ | X^n].$$

Lorden's formulation tries to minimize the worst possible detection delay subjected to a constraint on PFD [99]. It was shown in [97], that the exact optimal solution for Lorden's formulation of QCD can be obtained using the CumSum algorithm [100].

CumSum Algorithm

Define the statistic $W(n)$ such that

$$W(n) = \max_{1 \leqslant k \leqslant n+1} \sum_{i=k}^{n} \log(L(x_i)),$$

and $W_0 = 0$, where $L(X_n) = f_1(X_n)/f_0(X_n)$ is the log likelihood ratio. It can be shown that the statistic $W(n)$ has the following recursive form

$$W_{n+1} = (W_n + \log(L(X_{n+1}))^{+}).$$

Time at which a change occured (τ) is chosen such that

$$\tau_c = \inf\{n \geqslant 1 : W_n \geqslant b\},$$

where b is a threshold. More details about the CumSum algorithm can be found in [100], [96], [97], [95].

Proposed Algorithm

Based on the Bayesian and MiniMax QCD algorithms, we propose an authentication algorithm to detect intrusions in an AA system. Essentially, our proposal is independent of all other base elements of an AA system (Figure 7-3). Therefore, existing AA systems can easily be extended to incorporate the proposed QCD method.

Training: In the training phase, the user is asked to perform a wide variety of tasks and sensor data are obtained. Pre-determined features are then evaluated from the obtained sensor data. Part of the obtained features are stored in memory to serve as the gallery in the AA system. The remaining features are compared against chosen gallery to build a match distribution. In addition, the gallery entries are used to construct a non-match distribution based on the non-user features as illustrated in Figure 7-2. For the experiments conducted in this paper, a sample of other class data was used to model the non-match distribution. In practice, a common set of pre-obtained sensor data specific for the device can be used for this purpose. For example, face images of different users obtained from the same device can be made available in a cloud storage system for training.

```
input : Detection score of most recent iteration $score$, match score $x_n$,
        match distribution $f_0$, non-match distribution $f_1$, $Threshold$,
        $FloorThreshol$
output: Detection of an intrusion (Boolean)
//If it's the initial iteration set score to be zero;
if isempty(score) then
    $score = 0$ ;
else
    score = UpdateScore($score, x_n, f_0, f_1, FloorThreshold$);
    //FloorThreshold is used only resource efficient versions;
end
if $score > Threshold$ then
    Detect = True;
else
    Detect = False;
end
Return (Detect);
```

Algorithm 3: Main procedure proposed for decision making.

Testing: The proposed testing phase takes in to consideration a sequence of past observations when making a decision. At time n, the same set of sensor data and corresponding features g_n of the probe is collected as in the enrollment phase. Obtained features are compared against the signatures to obtain a score value x_n. A decision is made based on scores corresponding to all past observations $x_1, x_2, \cdots, x_n$ and the match distribution f_0 and non-match distribution f_1.

Described in Algorithm 3 is the proposed structure for decision making. A variable *score* is initialized at zero and is updated using the method *UpdateScore* once a new observation is observed. Once the score exceeds threshold A, a detection of a change is declared. In this paper we present two variants of the method *UpdateScore* based on BQCD and MQCD. Those methods are listed in Algorithm 4 and Algorithm 5, respectively.

```
input  : $score, x_n, f_0, f_1$
output : $score$
//Calculate likelihood ratio;
$L = f_1(x_n)/f_0(x_n)$;
$\tilde{p_n} = score + (1 - score)\rho$;
$score \leftarrow \frac{\tilde{p_n} L}{\tilde{p_n} L + (1 - \tilde{p_n})}$;
Return ($score$);
```

Algorithm 4: UpdateScore Method incorporating BQCD.

```
input  : $score, x_n, f_0, f_1$
output : $score$
//Calculate likelihood ratio;
$L = \log(f_1(x_n)/f_0(x_n))$;
$score \leftarrow score + L$;
Return ($score$);
```

Algorithm 5: UpdateScore Method incorporating MQCD.

Illustrated in Figure 7-5 is the variation of detection scores when Bayesian QCD is used for the video shown in Figure 7-1. Detection scores values increase when there is significant variation in the expression. However, they decrease again once the neutral expression is returned. Since the intrusion occurs in Frame 201, the score value is seemed to be monotonically increasing. In this specific example, the likelihood ratio becomes infinity after the change point. Therefore, according to Algorithm 4, the score is increasing by the assigned constant C. It should be noted that, slope of the curve could be increased by selecting a higher value for C in Algorithm 4. By the time the score passes the predetermined threshold, it is declared that an intrusion has occurred. For the set threshold in Figure 7-5, detection occurs with a delay of 9 samples.

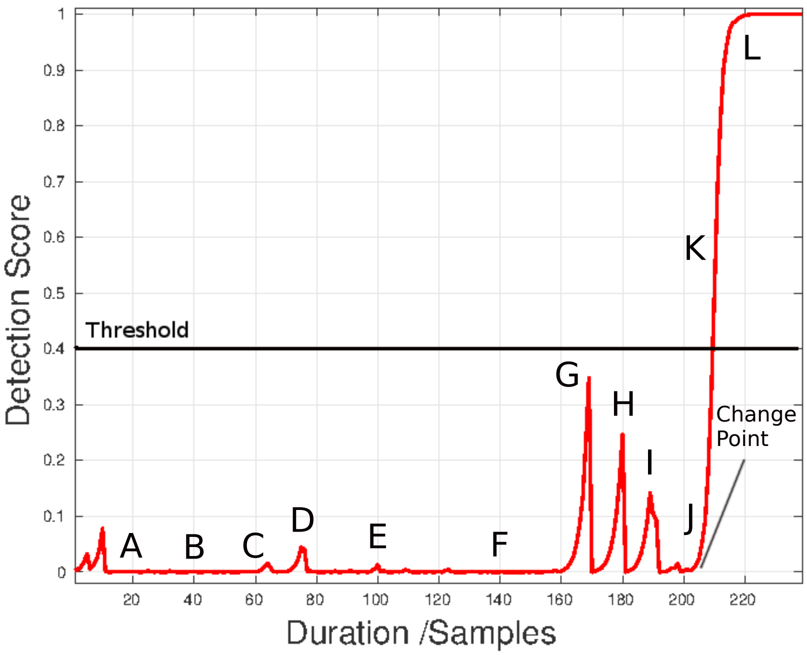


Figure 7-5. Variation of Bayesian QCD scores for the video shown in Figure 7-1.

Resource Efficient Quickest Change Detection

In this section, we discuss how QCD can be performed with having a lower burden on the device resources. As noted in the introduction, a trivial solution to the problem of resource limitation is to perform detection on a few selected samples of observations. However, quickest detection performance may degrade greatly depending on how the sampling is done. In what follows, we introduce a data driven sampling rule based on data efficient QCD [96], [97], [101].

Consider a sequence of time instances $t = 1, 2, \cdots, i$ in which the device operates. At each time $i, i > 0$, a decision is made whether to take or skip an observation at time $i + 1$. Let M_i be the indicator random variable such that $M_i = 1$ if the score x_i is used for decision making, and $M_i = 0$ otherwise. Thus, M_{i+1} is a function of the information available at time i, i.e. $M_{i+1} = \phi_i(I_i)$, where ϕ_i is the control law at time i, and $I_i = [M_1, M_2, \cdots, M_i, x_1^{M_1}, x_2^{M_2}, \cdots, s_i^{M_i}]$ represents the information at time i. Here, $x_i^{M_i}$ represents x_i if $M_i = 1$, otherwise x_i is absent from the information vector I_i. Let T be the stopping time on the information sequence $\{I_i\}$. Then, average percentage of observations (APO) obtained prior to the change point can be quantified as

$$APO = E\left[\frac{1}{T}\sum_{n=1}^{T} M_n\right]. \tag{7.4}$$

In the QCD scheme introduced in the previous section, observations are obtained at every time instant. Therefore, APO is equal to 1. A lower APO can be obtained while maintaining a lower ADD and PFD rates by employing an intelligent sampling mechanism. When such a mechanism is used, average usage of resources (memory, processing power, battery usage) are expected to decrease compared to the QCD scheme [**QCD_BTAS_2016**]. We introduce a technique to achieve this based on data efficient QCD.

Efficient Bayesian Formulation (E-BQCD)

In the Bayesian formulation of efficient QCD, an additional constraint based on the number of observations used is introduced in the optimization procedure. Define the Average Number of Observations (ANO) as

$$ANO = E\left[\sum_{n=1}^{min(\tau,T-1)} M_n\right]. \tag{7.5}$$

This quantity essentially captures the number of observations taken prior to the change point. It should be noted that ANO does not penalize additional observations taken after the change point. Therefore, ANO is a more conservative measure of the number of observations compared to APO, where $T \times APO \geq ANO$.

The efficient Bayesian QCD problem can be formulated as an optimization problem as follows[101],[97]

$$\begin{aligned} \underset{\phi,\tau}{\text{minimize}} \quad & ADD(\phi,\tau) \\ \text{subject to} \quad & PFA(\phi,\tau) \leq \alpha \\ & ANO(\phi,\tau) \leq \beta. \end{aligned} \tag{7.6}$$

In [101], an algorithm is presented to seek a possible solution for this optimization problem. Consider P_n, the probability that change had occurred by time n,

$$p_n = P(T \leq n | I_n),$$

where $p_0 = 0$. For $A, B \geq 0$ and $A > B$ the following control rule is proposed

$$M_{n+1} = \begin{cases} 0, & \text{if } p_n < B \\ 1, & \text{if } p_n \geq B. \end{cases}$$

Based on the value of M_{n+1}, p_{n+1} is updated as

$$p_{n+1} = \begin{cases} \tilde{p_n}, & \text{if } M_{n+1} = 0 \\ \frac{\tilde{p_n} L(x_{n+1})}{\tilde{p_n} L(x_{n+1}) + (1-\tilde{p_n})}, & \text{if } M_{n+1} = 1, \end{cases}$$

where $\tilde{p_n} = p_n + (1 - p_n)\rho$ and $L(x_{n+1}) = \frac{f_1(x_{n+1})}{f_0(x_{n+1})}$. An intruder detection is declared at the earliest time (τ_D) when p_n surpasses the threshold A, i.e. $\tau_D = inf\{n \geq 1 : p_n > A\}$. It was proved in [101] that this algorithm is asymptotically optimal for the optimization formulation (7.6) for each fixed β when $\alpha \to 0$.

Efficient MiniMax Formulation (E-MQCD)

In a non-Bayesian setting, due to the absence of a priori distribution on the change point, a different quantity should be used to quantify the number of observations used for decision making. Work in [96],[97], proposes change Duty Cycle (CDC) as

$$CDC = \lim_n \sup \frac{1}{n} E_n \left[\sum_{k=1}^{n-1} M_k | \tau \geq n \right] \tag{7.7}$$

for this purpose. It should be noted that both CDC and APO are similar quantities. With the definition of CDC, efficient QCD in a minimax setting can be formulated as the following optimization problem

$$\begin{aligned} \underset{\phi,\tau}{\text{minimize}} \quad & ADD(\phi,\tau) \\ \text{subject to} \quad & PFA(\phi,\tau) \leq \alpha \\ & CDC(\phi,\tau) \leq \beta. \end{aligned} \tag{7.8}$$

In [96], a two threshold algorithm called DE-CumSum algorithm, is presented as a solution to this optimization problem. For suitably selected thresholds chosen to meet constraints α and β, it is shown to obtain the optimal lower bound asymptotically as $\alpha \to 0$. The DE-CumSum algorithm is presented below.

Start with $W_0 = 0$ and let $\mu > 0, A > 0$ and $h \geq 0$. For $n \geq 0$ use the following control rule

$$M_{n+1} = \begin{cases} 0 & \text{if } W_n < 0 \\ 1 & \text{if } W_n \geq 0. \end{cases}$$

Statistic W_n is updated as follows

$$W_{n+1} = \begin{cases} min(W_n + \mu, o), & \text{if } M_{n+1} = 0 \\ max(W_n + logL(X_{n+1}), -h), & \text{if } M_{n+1} = 1, \end{cases}$$

where $L(x) = \frac{f_1(x)}{f_0(x)}$. A change is declared at time τ_W, when the statistic W_n passes the threshold A for the first time as

$$\tau_W = inf\{n \geq 1 : W_n > A\}. \tag{7.9}$$

Modified Algorithm

Testing and training procedure under the resource efficient QCD-based detection is the same as proposed in Section 7. Testing is done using the *main* method described in Algorithm 3 in section 7. Here, we present two alternative variants of the *UpdateScore* method based on resource efficient BQCD and MQCD. Different steps are summarized in Algorithm 6 and Algorithm 7, respectively corresponding to the updates of E-BQCD and E-MQCD.

In Algorithm 7.8, parameter D is a constant. In our tests, this parameter was set to be equal to 1.Parameter *FloorThreshold* is set equal to 0.05 in both algorithms.

```
input  : $score, x_n, f_0, f_1, FloorThreshold$
output : $score$
//Calculate the priori probability $L = f_1(x_n)/f_0(x_n)$;
$\tilde{p_n} = score + (1 - score)\rho$;
//Use priori to update score when score is small if $score < FloorThreshold$
 then
    $score = \tilde{p_n}$ ;
else
    $score \leftarrow \frac{\tilde{p_n} L}{\tilde{p_n} L + (1 - \tilde{p_n})}$ ;
end
Return ($score$);
```

Algorithm 6: UpdateScore method incorporating E-BQCD

```
input  : $score, x_n, f_0, f_1, FloorThreshold$
output : $score$
if $score < 0$ then
    $score$ =min(score+$D$,0) ;
else
    $score \leftarrow max(score + log(\frac{f_1(x_n)}{f_0(x_n)}), -FloorThreshold)$ ;
end
Return ($score$);
```

Algorithm 7: UpdateScore method incorporating E-MQCD

Evolution of score values when efficient Bayesian QCD is used is illustrated in Figure 7-6 for the case shown in Figure 7-1. In order to demonstrate the effect of using different sampling rates, the same experiment was conducted for a series of APO values. Functionality of efficient QCD algorithm can be explained using Figure 7-6. Consider the black line (corresponding to APO = 0.92%) in Figure 7-6. After the initial observation at $t = 1$, no observations are taken until the score passes 0.05 at $t = 52$. In this duration, score is updated using a priori probability. Hence, the score is having a constant slope in this interval. At $t = 52$, as the score passes 0.05, an observation is taken and the score is updated based on log-likelihood as outlined in Algorithm 6. This causes a discontinuity in the graph by shifting the value of score onto 0.0007. Since this value is lower than 0.05, no observation is taken at $t = 53$. This process is continued until the score value surpasses the $Threshold$ value when an intrusion is declared.

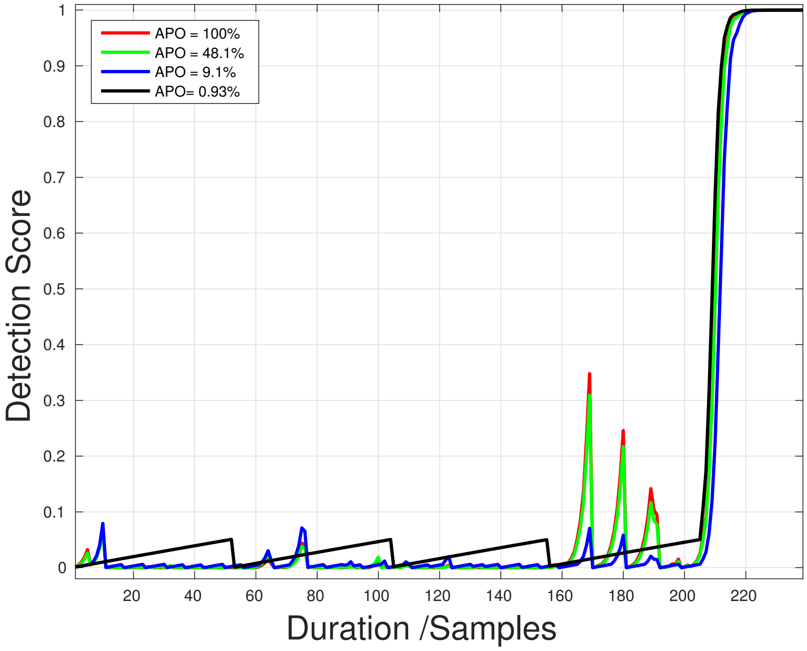


Figure 7-6. Variations of efficient Bayesian QCD scores for the video shown in Figure 7-1 for different APO values.

Furthermore, Figure 7-6 suggests that variations of scores across time is somewhat similar when APO is 100% and 48.1% for the considered case. This shows that selecting sampling points intelligently could reduce the sampling rate almost by half while producing near-identical performance for specific cases. The effect of sampling on the detection performance is discussed in detail in the following section.

Experimental Results

Quickest Change Detection

We evaluated the performance of the proposed QCD methods using three publicly available unconstrained AA datasets - Touchalytics [102], MOBIO [103], and UMDAA-01 [104]. The following three previously proposed AA methods are used as the benchmark for comparisons.

Single score-based authentication (SSA): The present score value x_n alone is used to authenticate the user. If the score value is above a predetermined threshold, user is authenticated otherwise treated as an intrusion.

Time decay fusion (Sui et al.) [105]: In this method, two score samples fused by a linear function is used along with a decaying function to determine the authenticity of a user as, $s_n = wx_{n-1} + (1-w)x_n \times e^{\tau\delta t}$, where, w, τ are constants and δt is the time elapsed since the last observation.

Confidence functions (Crouse et al.) [106]: A sequential detection score S_{login} is calculated by incorporating time delay since the last observation and a function of the present score x_n. The detection score is evaluated as, $S_{login,n} = S_{login,n-1} + f_{map}(x_n) + \int_{t_{prev}}^{t_{now}} f_{dec} dt$. See [106] for the exact definitions of f_{map} and f_{dec}.

The PFD-ADD curves, introduced in Section 7, are used to compare the performance of different methods. The PFD-ADD plot for the BQCD and MQCD methods can be obtained by varying the parameter *Threshold* and plotting the ADD values corresponding to different PFD values. Similarly, the ADD-PFD curves for SSA and the method proposed by Sui et al. [105] and Crouse et al. [106] are obtained by varying the decision making threshold.

The measure of ADD signifies the latency of detecting an attack. On the other hand, PFD is a measure of false detections. A practical AA system should have a low latency in decision making as well as low false detection rate. Therefore, better AA systems are expected to have low ADD and PFD values. Hence, they should operate towards the lower left corner of the PFD-ADD curve, as illustrated in Figure 7-4. As a result, AA methods with very low operating values in the PFD-ADD plot are better in terms of their performance.

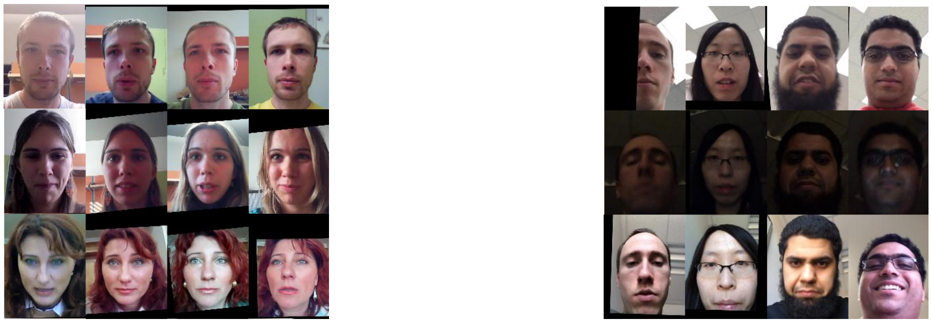

(a) (b)

Figure 7-7. Sample detected face images from (a) the MOBIO dataset and (b) the UMDAA-01 dataset.

In the absence of a proper mobile dataset with intrusions, experimental data was obtained in the following manner for all datasets considered. For each dataset, all possible pairs of users were considered at a time. For each pair of users, full length signals (e.g. touch gestures or detected faces) of considered pair of users were merged to obtain a trial with a single intrusion. As a result, only one intruder/attacker was presented at each trial. Shown in Figure 7-1 is a sample trial obtained in this manner. Frames A to I correspond to the enrolled images of the genuine user. An intruder is presented at frame J and onwards. The intrusion point depends on the length of the samples corresponding to the first (genuine) user and therefore is not pre-determined. Each trial was tested using before mentioned methods to determine detection delay and probability of false detections under each method.

UMDAA-01 Dataset

The UMDAA-01 dataset [104] consists of images of 50 individuals taken from an iPhone 5 device across three sessions performing five tasks including an enrollment task. Both face images as well as touch gestures are simultaneously captured in this dataset. Sample detected face images from this dataset are shown in Figure 7-7(b). As suggested in [104], enrollment data was used as gallery and data from the other sessions was used as probes. In addition 20 number of instances from the probe session was used to obtain the match score distribution. When testing, 33 % of the remaining subjects excluding the probe class and the target class were randomly chosen to obtain the non-match distribution.

Results on the Face Data: Face images of the user were normalized and image regions corresponding to eyes, nose, lips and eyebrows were extracted. The HOG features [107] were extracted on each facial component. These features were concatenated to obtain the resulting feature for the given face. Cosine distance is used to generate score values by matching enrollment data with probes. Figure 7-8 shows the ADD-PFD plot corresponding the UMDAA-01 face data. From this figure, it can be seen that both BQCD ($\rho = 0.001$) and MQCD outperform the other methods. This can be seen by comparing their performances in the low PFD region.

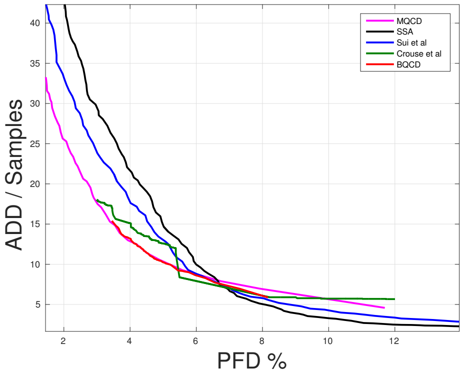


Figure 7-8. Performance curves obtained on the UMDAA-01 face dataset.

Results on the Touch Data: From each swipe data, a 27-dimensional feature vector is extracted using the method described in [102]. A single class SVM with RBF kernel was used to generate matching scores. Figure 7-9 shows the ADD-PFD curves corresponding to different methods on this dataset. It should be noted that there exists a considerable similarity between single touch swipes of different users. Therefore, from Figure 7-9, methods that rely on data of single or two swipes have performed poorly. It can be seen that BQCD, MQCD and the method proposed by Crouse et al. [106] that uses information from pre and post change distributions have performed reasonably well. In general, the MQCD method yields faster detection rates and low false detections compared to the other methods.

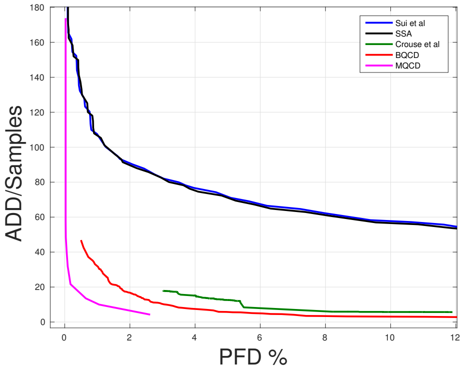


Figure 7-9. Performance curves obtained on the UMDAA-01 touch dataset.

MOBIO Dataset

The MOBIO dataset [103] contains videos of 152 subjects taken across two phases where each phase consists of six sessions each. Videos in this dataset are acquired using a standard 2008 Macbook laptop computer and a NOKIA N93i mobile phone (See Figure 7-7(a)). Following the protocol defined in [108], video frames of the 12th session were considered as the enrollment data and video frames of all other sessions were used as probes. We conducted our experiments on the laptop image data based on the LBP features. Again, the cosine distance was used to generate the match and non-match scores. Figure 7-10 shows the performance curves corresponding to different methods on the MOBIO dataset. Note that the images in this dataset are well aligned and mostly frontal. As a result, pre-change and post-change distributions are well separated. Hence, all considered methods yielded relatively better performance. However, the BQCD and MQCD methods have performed marginally better than the other compared approaches.

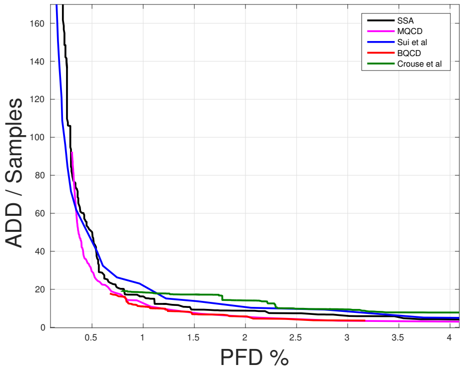


Figure 7-10. Performance curves obtained on the MOBIO face dataset.

Touchalytics Dataset

The Touchalitics dataset contains touch data of 37 users collected across 7 tasks. Similar to the UMDAA-01 touch dataset, touch gesture features are extracted using the method described in [102] and a single class SVM with RBF kernel was used to generate match and non-match scores. Figure 7-11 shows the performance of different methods on this dataset. As before, making a decision based on a single swipe or two swipes have appeared to perform poorly. The MQCD method performs the best followed by the BQCD method and the method of Crouse et al. [106].

Discussion

From the above experiments, it can be seen that the BQCD and MQCD methods have outperformed the other existing AA methods. Furthermore, in all cases, the MQCD method has performed marginally better than the BQCD method. This is mainly due to the error induced by approximating the change distribution by a Geometric (ρ) distribution. In practice, where information on change (intrusion) probability is

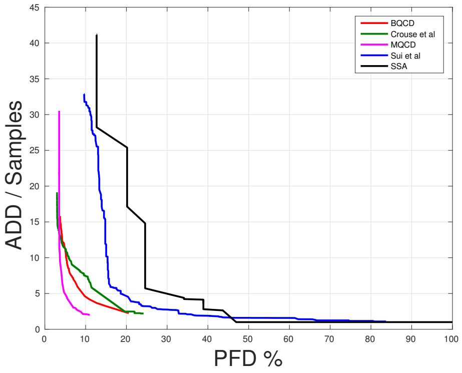


Figure 7-11. Performance curves obtained on Touchalytics dataset.

unknown in advance, the MQCD method provides more usability as opposed to the BQCD method.

Detection delay and probability of false detections of the proposed algorithm depend on the type of features as well as the classifiers used for matching. The proposed method is not restricted to any specific type of feature or a classifier. Therefore, by using better features and classifiers it is possible to obtain even lower ADD and PFD values.

Furthermore, it should be noted that, the detection delay rates (ADD) shown in Figures 7-8, 7-9, 7-10, and 7-11 are highly inflated as a result of non-detected intrusions due to the limitations of the features and/or classifiers. To further elaborate on this point, let us consider the implementation of the MQCD method with a threshold chosen such that PFD is at 5%. Tabulated in Table 7-III is the distribution of detection delay (ADD) for the tests conducted. According to Table 7-III, nearly 90% of the time, an intrusion can be detected using less than 7 samples. Therefore, the proposed method would produce quick results for a small false detection rate in a practical setting.

	2-3 S	4-5 S	6-7 S	8-10S	>10 S
UMD-Face	11.9	17.07	55.06	6.02	9.93
UMD-Touch*	73.62	13.51	4.69	3.04	3.13
MOBIO	8.74	61.87	10.38	7.51	11.5
Touchalytics	3.65	7.23	82.23	2.94	3.94
Mean	24.47	24.92	38.09	4.87	7.12

Table 7-III. Percentage breakdown of delay times (in samples) for a fixed PFD of 5% for MQCD. *3% of PFD was used instead.

Resource Efficient QCD

Effect of extending QCD to incorporate resource efficiency through sampling was studied on the before mentioned three datasets. Performance of the proposed sampling method was compared against the following two benchmark sampling methods.

Fixed Time Step Sampling: Most of the existing AA systems employ a sampling mechanism where sensor observations are obtained with a fixed inter-sample interval [106],[105]. In our experiments, this interval was chosen to satisfy the given APO rate.

Dice Sampling: In this method, a weighted coin is tossed at every time instant to determine whether a sample should be obtained or not [101],[96],[97]. The weight of the coin is equal to the chosen APO value.

Same set of features and classifiers as described in Section 7 were used to evaluate performance of the proposed methods. For each dataset considered in this paper, E-BQCD and E-MQCD were applied on top of BQCD and MQCD for a specific APO rate. In addition, BQCD method was implemented using time step sampling and DICE sampling for comparison.

Shown in Figure 7-12 are the performance curves obtained for the UMDAA-01 face dataset for an APO of 21%. Performance curves have shifted to the left by some margin and have moved slightly upwards as shown in the graph due to sampling. At

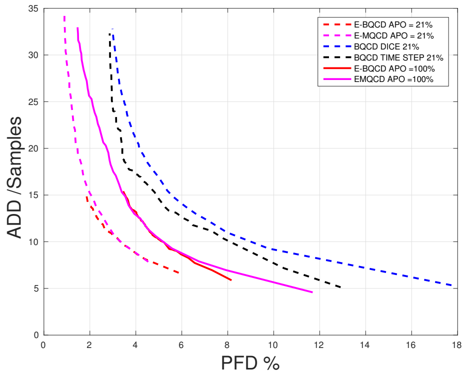


Figure 7-12. Performance curves obtained on the UMDAA-01 face dataset for efficient QCD.

a glance, performance appears to have improved despite lower sampling for a given PFD value. In comparison, sampling with DICE and fixed time step has worsen the initial result. The same trend seems to follow in the UMDAA-01 touch dataset as seen in Figure 7-13 for an APO of 17%. Although BQCD tends to perform poorly compared to MQCD, performance of resource efficient versions of BQCD and MQCD are comparable.

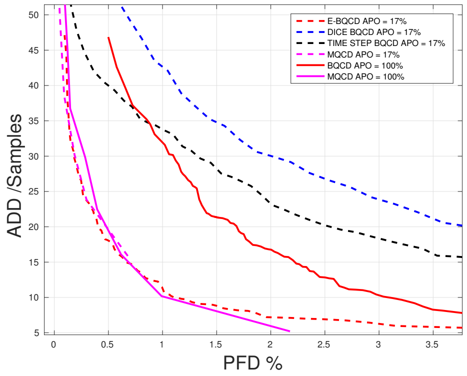


Figure 7-13. Performance curves obtained on the UMDAA-01 touch dataset for efficient QCD.

Results obtained for the experiments done on the MOBIO face dataset for an APO of 17% are shown in Figure 7-14. Both QCD methods yielded comparable results on the MOBIO dataset in our earlier experiments. When resource efficient QCD was employed, performance curves for both methods improved nearly by an equal amount compared to the QCD performance. It should be noted that, both DICE and fixed time sampling performances are much worse compared to E-QCD on the MOBIO dataset.

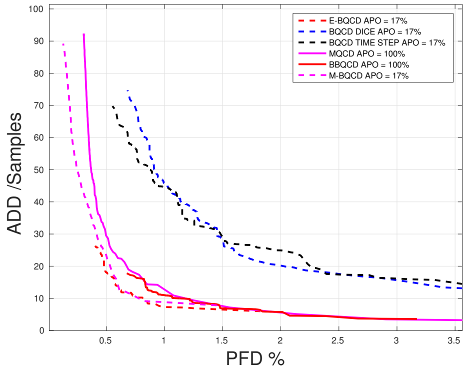


Figure 7-14. Performance curves obtained on the MOBIO face dataset for efficient QCD.

Final set of experiments were carried out on the Touchalytics touch dataset with an APO rate of 17%. Results of these experiments are presented in Figure 7-15. As in earlier cases, resource efficient QCD has outperformed QCD and other sampling methods. However, there are a couple of notable differences. Unlike in earlier experiments, E-BQCD and E-MQCD performance curves do not overlap in this case. However, this is only due to the absence of a common operating region. In addition, time step sampling performed better than DICE sampling on this dataset.

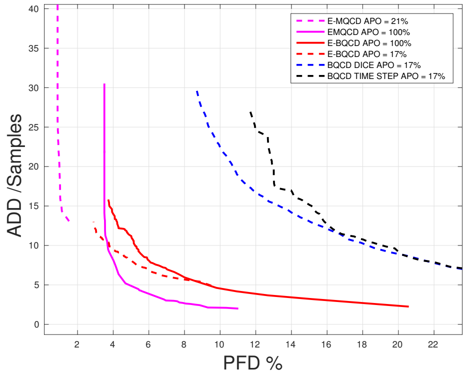


Figure 7-15. Performance curves obtained on the Touchalytics touch dataset for efficient QCD.

Resource efficient QCD have improved the performance of QCD and has performed better than alternative sampling methods have. The exact shape of the performance curves and gaps between each curves depend on the type of feature and classifier used. Irrespective of this, efficient QCD has yielded better results on average. This can be seen from the results summarized in Table 7-IV, where PFD values obtained for a fixed detection delay of 15 samples for all considered datasets are listed.

Table 7-IV. The PFD % rate for a detection delay of 15 samples. The efficient QCD methods use an APO of 17%. * APO of 21% used.

	SSA	Sui et al [105]	Crouse et al [106]	BQCD	MQCD	E-BQCD	E-MQCD	DICE [97]	TIMESTEP [105],[106]
UMD-Face*	5.1	4.6	4.0	3.5	3.4	**1.8**	2.1	5.5	4.8
UMD-Touch	48.8	50.2	4.0	2.2	**0.6**	**0.6**	**0.6**	5.4	3.9
MOBIO	1.1	1.2	1.7	0.8	0.8	**0.5**	0.6	3.0	3.5
Touchalytics	24.6	14.8	3.1	3.8	3.5	NA	**1.0**	13.1	15.0

In order to investigate this phenomena further, we carried out a case study on the UMDAA-01 face dataset. We conducted the above mentioned experiment on the dataset for a range of APO values. The resulted performance curves are shown in Figure 7-16. Performance curves in Figure 7-16 suggest that as APO decreases, performance curves keep on shifting further left. However, at the same time, the minimum possible detection time has also increased. Therefore, very low sampling is not feasible if quick change detection is desired. On the other hand, for a fixed detection delay, it might be possible to select a lower sampling rate so that lower PFD is obtained. This result is true for all datasets we considered as evident from Table 7-IV.

In Figure 7-17, we plot the minimum possible detection time for different APO values for the test conducted on UMDAA-01 face dataset. As evident from this figure, the minimum detection times increase as sampling fraction (APO) is increased. Therefore, for practical applications, it is desired to select a moderate value for APO when efficient QCD is used.

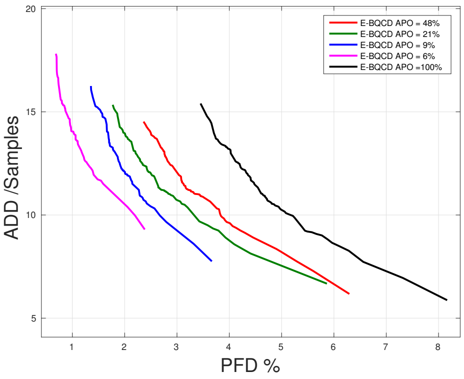


Figure 7-16. Effect of using different APO values for sampling.

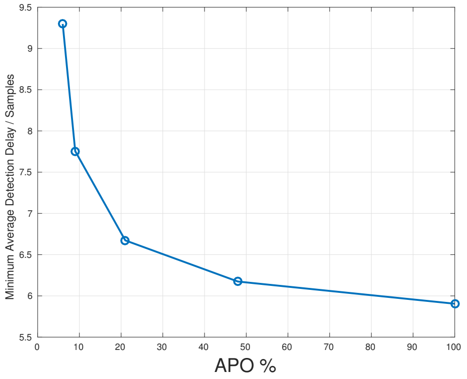


Figure 7-17. Effect of sampling on minimum average detection delay.

Discussion

Decreasing the sampling rate by selecting a threshold to achieve a lower APO have decreased the PFD while increasing the ADD for all the cases considered. This observation can be justified based on the score update mechanism. Consider the black curve (APO = 0.93%) and the blue curve (APO = 9.1 %) in the score evolution shown in Figure 7-6. Note that, with the way how sampling is performed, the black curve had missed all the humps created due to irregularities in sensing. The blue curve on the other hand, is affected only by a few humps due to sampling. However, due to low sampling in humps, the blue curve has not risen as much as the red curve (APO = 100%) has. This suggests that for a constant threshold, occurrences of false detections will be lower for the blue curve (APO = 9.1 %). Therefore, PFD decreases when sampling is carried out. On the other hand, due to sampling, an intrusion may not be sensed till the sampling that follows is carried out. This is clearly seen in the case of the blue line in Figure 7-6. As a result, detection delay increases for a fixed threshold when sampling is carried out as shown in Figure 7-17.

Summary

In this chapter, we addressed the issue of detecting novel samples with quickest time with high utilization of resources. We considered Active Authentication as an application for this discussion . We presented a method for detecting an intrusion in an AA system with a minimal delay with a constraint on false detection rate. Two variants of the QCD based on Bayesian and MiniMax formulations were introduced. Performance of the proposed method was demonstrated using three publicly available datasets.

The basic QCD methodologies were extended using resource efficient QCD where a data driven observation sampling was introduced with the aim of increasing resource efficiency. The introduced algorithms not only reduced number of observations taken, but also improve the performance of the system in terms of latency and false detections. Validity of this result was demonstrated using various datasets.

The proposed method does not rely on a specific feature or a classifier for its performance. This was verified in testing by using different classifiers and features for different datasets. Therefore, existing AA methods can be extended using the proposed method to enhance the performance. It was shown that the proposed method is effective even when there is a considerable overlap between pre and post-change distributions.

Chapter 8

Discussion and Future Work

In this thesis, we considered the problem of novelty detection in four different settings. For each setting, a deep-learning based solution was presented and the effectiveness of the method was assessed on multiple benchmark datasets.

First, we propose a transfer learning-based solution for the problem of *Multi-class novelty detection with out of distribution data*. In particular, we proposed an end-to-end deep-learning based approach in which we investigate how the knowledge contained in an external, out-of-distributional (OOD) dataset can be used to improve the performance of a deep network for visual novelty detection. Our solution differs from the standard deep classification networks on two accounts. First, we use a novel loss function, *membership loss*, in addition to the classical cross-entropy loss for training networks. Secondly, we used the knowledge from the external dataset more effectively to learn *globally negative filters*, filters that respond to generic objects outside the known class set. We showed that thresholding the maximal activation of the proposed network can be used to identify novel objects effectively.

We extended this solution for *one-class novelty detection with OOD data* where labeled out-of-distributional is used for feature learning in one-class classification. This method operates on top of a Convolutional Neural Network (CNN) of choice and produces descriptive features while maintaining a low intra-class variance in the feature space for the given class. For this purpose two loss functions, *compactness loss* and *descriptiveness loss* are used along with a parallel CNN architecture. A template matching-based framework is introduced to facilitate the testing process.

In addition, the problem of *open-set recognition*, where the goal is to determine if a given sample belongs to one of the classes used for training a model (known classes) was studied. The main challenge in open-set recognition is to disentangle open-set samples that produce high class activations from known-set samples. Two techniques are presented to force class activations of open-set samples to be low. First, a generative model is trained for all known classes and then the input is augmented with the representation obtained from the generative model to learn a classifier. This network learns to associate high classification probabilities both when the image content is from the correct class as well as when the input and the reconstructed image are consistent with each other. Second, self-supervision was used to force the network to learn more informative features when assigning class scores to improve separation of classes from each other and from open-set samples.

Then, the classical problem of *one-class novelty detection* was considered. Given a set of examples from a particular class, the goal is to determine if a query example is from the same class. Presented method is based on learning latent representations of in-class examples using a denoising auto-encoder network. The key contribution of the algorithm is to explicitly constrain the latent space to *exclusively* represent the given class. In order to accomplish this goal, firstly, latent space is forced to have bounded support by introducing a *tanh* activation in the encoder's output layer. Secondly, using a discriminator in the latent space that is trained adversarially, encoded representations of in-class examples are ensured to resemble uniform random samples drawn from the same bounded space. Thirdly, using a second adversarial discriminator in the input space, is it made sure that all randomly drawn latent samples generate examples that look real. Finally, a gradient-descent based sampling technique that explores points in the latent space that generate potential out-of-class examples is used accelerate the training process.

Finally, We discussed challenges arising in a practical dynamic novelty detection system with respect to latency and data efficiency. We considered Active Authentication (AA) – which is a practical application of novelty detection as a case study for this purpose. Quickest change detection and its data efficient extension were discussed as potential solutions to this problem. We demonstrated how these solutions can be used to obtain decisions with low latency in a resource constrained AA system.

In the future, I hope to address challenges practical novelty detection schemes come across. First, I will study how deep learning based novelty detection can be applied in large scale classification systems. In our experiments we observed that models performing well in smaller datasets doesn't necessary scale well to larger datasets such as ImageNet. I hope to fuse predictions of multiple networks to arrive at more robust predictions. In addition, I will study how novelty detectors can defend against adversarial attacks. I will study different attack models for novelty detection and present defense mechanisms to defend against them.

Further, I hope to study how deep learning-based novelty detection can be deployed across different domains. In particular, I will study how a novelty detector trained on a given domain can be adapted to a second domain when only few annotated samples are available from the second domain. In addition, I will study how privacy preserving deep novelty detection can be carried on mobile devices using Federated Learning [109]. Finally, I will study how novelty detection can be applied in applications such as video surveillance and biometric anti-spoofing.

References

1. Bendale, A. & Boult, T. *Towards Open World Recognition* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015).
2. Markou, M. & Singh, S. Novelty detection: a review – Part 1: statistical approaches. *Signal Processing* **83,** 2481–2497 (2003).
3. Chandola, V., Banerjee, A. & Kumar, V. Anomaly Detection: A Survey. *ACM Comput. Surv.* **41,** 15:1–15:58 (2009).
4. Roberts, S. J. Novelty detection using extreme value statistics. *IEE Proceedings-Vision, Image and Signal Processing* **146,** 124–129 (1999).
5. Patel, V. M., Chellappa, R., Chandra, D. & Barbello, B. Continuous User Authentication on Mobile Devices: Recent progress and remaining challenges. *IEEE Signal Processing Magazine* **33,** 49–61 (July 2016).
6. Oza, P. & Patel, V. M. *Active Authentication using an Autoencoder regularized CNN-based One-Class Classifier* in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)* (2019).
7. Perera, P. & Patel, V. M. *Towards Multiple User Active Authentication in Mobile Devices* in *IEEE International Conference on Automatic Face and Gesture Recognition* (2017).
8. Mahbub, U., Sakar, S., Patel, V. & Chellappa, R. *Active authentication for smartphones: A challenge data set and benchmark results* in *IEEE International Conference on Biometrics: Theory, Applications and Systems* (Sept. 2016).
9. Scheirer, W. J., Rocha, A., Sapkota, A. & Boult, T. E. Towards Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)* **36** (7 July 2013).
10. Vyas, A. *et al. Out-of-Distribution Detection Using an Ensemble of Self Supervised Leave-Out Classifiers* in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII* (2018), 560–574.
11. Liu, J., Lian, Z., Wang, Y. & Xiao, J. *Incremental Kernel Null Space Discriminant Analysis for Novelty Detection* in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), 4123–4131.
12. Ruff, L. *et al. Deep One-Class Classification* in *Proceedings of the 35th International Conference on Machine Learning* (2018), 4393–4402.
13. Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)* (2006).

14. Hoffmann, H. Kernel PCA for Novelty Detection. *Pattern Recognition* **40,** 863–874 (2007).

15. Hadsell, R., Chopra, S. & Lecun, Y. *Dimensionality reduction by learning an invariant mapping* in *In Proc. Computer Vision and Pattern Recognition Conference (CVPR'06* (IEEE Press, 2006).

16. Sakurada, M. & Yairi, T. *Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction* in *Proceedings of the MLSDA 2014 2Nd Workshop on Machine Learning for Sensory Data Analysis* (2014).

17. Sabokrou, M., Khalooei, M., Fathy, M. & Adeli, E. *Adversarially Learned One-Class Classifier for Novelty Detection* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 3379–3388.

18. Goodfellow, I. *et al. Generative adversarial nets* in *Advances in neural information processing systems* (2014), 2672–2680.

19. Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U. & Langs, G. *Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery* in *Ipmi* (2017).

20. Schölkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J. & Williamson, R. C. Estimating the Support of a High-Dimensional Distribution. *Neural Comput.* **13,** 1443–1471 (2001).

21. Tax, D. M. J. & Duin, R. P. W. Support Vector Data Description. *Mach. Learn.* **54,** 45–66 (2004).

22. Pidhorskyi, S., Almohsen, R. & Doretto, G. in *Advances in Neural Information Processing Systems 31* (eds Bengio, S. *et al.*) 6822–6833 (Curran Associates, Inc., 2018).

23. Abati, D., Porrello, A., Calderara, S. & Cucchiara, R. *Latent Space Autoregression for Novelty Detection* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019).

24. Perera, P. & Patel, V. Learning Deep Features for One-Class Classification. *IEEE Transactions on Image Processing* **28,** 5450–5463 (May 2019).

25. Rodner, E., Wacker, E.-s., Kemmler, M. & Denzler, J. *One-class classification for anomaly detection in wire ropes with gaussian processes in a few lines of code* in *In: Conference on Machine Vision Applications (MVA)* (2011), 219–222.

26. Rodner, E. *et al.* Maximally Divergent Intervals for Anomaly Detection. *CoRR* **abs/1610.06761** (2016).

27. Clifton, D. A., Hugueny, S. & Tarassenko, L. Novelty detection with multivariate extreme value statistics. *Journal of signal processing systems* **65,** 371–389 (2011).

28. Bodesheim, P., Freytag, A., Rodner, E., Kemmler, M. & Denzler, J. *Kernel Null Space Methods for Novelty Detection* in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2013).

29. Dufrenois, F. & Noyer, J. *A null space based one class kernel Fisher discriminant* in *International Joint Conference on Neural Networks (IJCNN)* (2016).

30. Markou, M. & Singh, S. Novelty detection: a review – Part 2: neural network based approaches. *Signal Processing* **83,** 2499–2521 (2003).

31. Samangouei, P. & Chellappa, R. *Convolutional neural networks for attribute-based active authentication on mobile devices* in *IEEE International Conference on Biometrics: Theory, Applications and Systems* (Sept. 2016).

32. Krizhevsky, A., Sutskever, I. & Hinton, G. E. *ImageNet Classification with Deep Convolutional Neural Networks* in *Advances in Neural Information Processing Systems 25* (2012), 1097–1105.

33. Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* **abs/1409.1556** (2014).

34. Chopra, S., Hadsell, R. & LeCun, Y. *Learning a similarity metric discriminatively, with application to face verification* in *IEEE Conference on Computer Vision and Pattern Recognition* **1** (June 2005), 539–546.

35. Donahue, J. *et al. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition* in *Proceedings of the 31st International Conference on Machine Learning* (eds Xing, E. P. & Jebara, T.) **32** (Bejing, China, 22–24 Jun 2014), 647–655.

36. Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. *Extracting and Composing Robust Features with Denoising Autoencoders* in *Proceedings of the 25th International Conference on Machine Learning* (2008), 1096–1103.

37. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. & Manzagol, P.-A. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research* **11,** 3371–3408 (2010).

38. Kingma, D. P. & Welling, M. *Auto-encoding variational Bayes* in *International Conference on Learning Representations* (2014).

39. Perera, P. & Patel, V. M. *Deep Transfer Learning for Multiple Class Novelty Detection* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019).

40. Hein, M., Andriushchenko, M. & Bitterwolf, J. *Why ReLU Networks Yield High-Confidence Predictions Far Away From the Training Data and How to Mitigate the Problem* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019).

41. Bendale, A. & Boult, T. *Towards Open Set Deep Networks* in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on* (2016).

42. Ge, Z., Demyanov, S. & Garnavi, R. *Generative OpenMax for Multi-Class Open Set Classification* in *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017* (2017).

43. Neal, L., Olson, M., Fern, X., Wong, W.-K. & Li, F. *Open Set Learning with Counterfactual Images* in *The European Conference on Computer Vision (ECCV)* (Sept. 2018).

44. Oza, P. & Patel, V. M. *C2AE: Class Conditioned Auto-Encoder for Open-Set Recognition* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019).

45. Yoshihashi, R. *et al. Classification-Reconstruction Learning for Open-Set Recognition* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019).

46. Rasmus, A., Berglund, M., Honkala, M., Valpola, H. & Raiko, T. in *Advances in Neural Information Processing Systems 28* (eds Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M. & Garnett, R.) 3546–3554 (Curran Associates, Inc., 2015).

47. Turk, M. & Pentland, A. Eigenfaces for recognition. *Journal of Cognitive Neuroscience* **3,** 71–86 (1991).

48. Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S. & Ma, Y. Robust Face Recognition via Sparse Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **31,** 210–227 (Feb. 2009).

49. Bendale, A. & Boult, T. E. *Towards Open Set Deep Networks* in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016* (June 2016), 1563–1572.

50. Schultheiss, A., Käding, C., Freytag, A. & Denzler, J. *Finding the Unknown: Novelty Detection with Extreme Value Signatures of Deep Neural Activations* in *Pattern Recognition - 39th German Conference, Proceedings* (2017), 226–238.

51. Bodesheim, P., Freytag, A., Rodner, E. & Denzler, J. *Local Novelty Detection in Multi-class Recognition Problems* in *2015 IEEE Winter Conference on Applications of Computer Vision* (2015), 813–820.

52. Oza, P. & Patel, V. M. One-Class Convolutional Neural Network. *IEEE Signal Processing Letters* **26,** 277–281 (2019).

53. Mirza, M. & Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

54. Radford, A., Metz, L. & Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *CoRR* **abs/1511.06434.** arXiv: `1511.06434` (2015).

55. Doersch, C., Gupta, A. & Efros, A. A. *Unsupervised Visual Representation Learning by Context Prediction* in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)* (2015), 1422–1430.

56. Doersch, C. & Zisserman, A. *Multi-Task Self-Supervised Visual Learning* in *The IEEE International Conference on Computer Vision (ICCV)* (Oct. 2017).

57. Gidaris, S., Singh, P. & Komodakis, N. Unsupervised Representation Learning by Predicting Image Rotations. *ArXiv* **abs/1803.07728** (2018).

58. Golan, I. & El-Yaniv, R. in *Advances in Neural Information Processing Systems 31* (eds Bengio, S. *et al.*) 9758–9769 (Curran Associates, Inc., 2018).

59. He, K., Zhang, X., Ren, S. & Sun, J. *Deep Residual Learning for Image Recognition* in *IEEE Conference on Computer Vision and Pattern Recognition* (June 2016), 770–778.

60. Russakovsky, O. *et al.* ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vision* **115,** 211–252 (Dec. 2015).

61. Yosinski, J., Clune, J., Nguyen, A., Fuchs, T. & Lipson, H. *Understanding Neural Networks Through Deep Visualization* in *Deep Learning Workshop, International Conference on Machine Learning (ICML)* (2015).

62. Huang, G., Liu, Z., van der Maaten, L. & Weinberger, K. Q. *Densely connected convolutional networks* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017).

63. Deng, J. *et al. ImageNet: A Large-Scale Hierarchical Image Database* in *Cvpr09* (2009).

64. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A. & Torralba, A. Places: A 10 million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

65. Krizhevsky, A., Nair, V. & Hinton, G. CIFAR-10 (Canadian Institute for Advanced Research).

66. Girshick, R., Donahue, J., Darrell, T. & Malik, J. *Rich feature hierarchies for accurate object detection and semantic segmentation* in *Computer Vision and Pattern Recognition* (2014).

67. Belhumeur, P. N., Hespanha, J. P. & Kriegman, D. J. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19,** 711–720 (1997).

68. Van der Maaten, L. & Hinton, G. E. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research* **9,** 2579–2605 (2008).

69. Gubner, J. A. *Probability and Random Processes for Electrical and Computer Engineers* (2006).

70. Jia, Y. *et al. Caffe: Convolutional Architecture for Fast Feature Embedding* in *Proceedings of the 22Nd ACM International Conference on Multimedia* (Orlando, Florida, USA, 2014), 675–678.

71. Chalapathy, R., Menon, A. K. & Chawla, S. Anomaly Detection using One-Class Neural Networks. *CoRR* (2018).

72. Saleh, B., Farhadi, A. & Elgammal, A. *Object-Centric Anomaly Detection by Attribute-Based Reasoning* in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition* (2013), 787–794.

73. Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. & Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* **88,** 303–338 (2010).

74. Nilsback, M.-E. & Zisserman, A. *Automated Flower Classification over a Large Number of Classes* in *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing* (Dec. 2008).

75. LeCun, Y. & Cortes, C. MNIST handwritten digit database.

76. Xiao, H., Rasul, K. & Vollgraf, R. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv: `cs.LG/1708.07747 [cs.LG]` (Aug. 28, 2017).

77. Nene, S. A., Nayar, S. K. & Murase, H. *Columbia Object Image Library (COIL-20* tech. rep. (1996).

78. Van den Oord, A. *et al.* in *Advances in Neural Information Processing Systems 29* (eds Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I. & Garnett, R.) 4790–4798 (2016).

79. Hendrycks, D. & Gimpel, K. *A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks* in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (2017).

80. Golan, I. & El-Yaniv, R. in *Advances in Neural Information Processing Systems 31* (eds Bengio, S. *et al.*) 9758–9769 (Curran Associates, Inc., 2018).

81. Xia, Y., Cao, X., Wen, F., Hua, G. & Sun, J. *Learning Discriminative Reconstructions for Unsupervised Outlier Removal* in *The IEEE International Conference on Computer Vision (ICCV)* (Dec. 2015), 1511–1519.

82. Zagoruyko, S. & Komodakis, N. *Wide Residual Networks* in *Bmvc* (2016).

83. Arjovsky, M., Chintala, S. & Bottou, L. *Wasserstein Generative Adversarial Networks* in *Proceedings of the 34th International Conference on Machine Learning* (eds Precup, D. & Teh, Y. W.) **70** (Pmlr, International Convention Centre, Sydney, Australia, June 2017), 214–223.

84. Netzer, Y. *et al.* in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning* (2011).

85. Krizhevsky, A., Nair, V. & Hinton, G. CIFAR-100 (Canadian Institute for Advanced Research).

86. Yu, F., Zhang, Y., Song, S., Seff, A. & Xiao, J. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. *arXiv preprint arXiv:1506.03365* (2015).

87. Liang, S., Li, Y. & Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. *International Conference on Learning Representations (ICLR)* (2018).

88. Van der Maaten, L. & Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **9,** 2579–2605 (2008).

89. Khan, H., Hengartner, U. & Vogel, D. *Usability and Security Perceptions of Implicit Authentication: Convenient, Secure, Sometimes Annoying* in *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)* (2015), 225–239.

90. Clarke, N., Karatzouni, S. & Furnell, S. in (eds Gritzalis, D. & Lopez, J.) 1–12 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2009).

91. Crawford, H. & Renaud, K. Understanding user perceptions of transparent authentication on a mobile device. *Journal of Trust Management* **1,** 1–28 (2014).

92. Lee, W. *Mobile Apps and Power Consumption - Basics, Part 1* https://developer.qualcomm.com/blog/mobile-apps-and-power-consumption-basics-part-1 (2016).

93. Samangouei, P., Patel, V. M. & Chellappa, R. Facial attributes for active authentication on mobile devices. *Image and Vision Computing* (May 2016).

94. Braz, C. & Robert, J.-M. *Security and Usability: The Case of the User Authentication Methods* in *Proceedings of the 18th Conference on L'Interaction Homme-Machine* (2006), 199–203.

95. Veeravalli, V. V. & Banerjee, T. Quickest Change Detection. *ArXiv e-prints.* arXiv: `1210.5552` (Oct. 2012).

96. Banerjee, T. & Veeravalli, V. Data-Efficient Quickest Change Detection in Minimax Settings. *IEEE Transactions on Information Theory,* 6917–6931 (Oct. 2013).

97. Banerjee, T. & Veeravalli, V. V. Data-Efficient Quickest Change Detection. *Sri Lankan Journal of Applied Statistics, Special Issue: Modern Statistical Methodologies in the Cutting Edge of Science,* 183–208 (Nov. 2014).

98. Shiryaev, A. N. On Optimum Methods in Quickest Detection Problems. *Theory of Probability & Its Applications* **8,** 22–46 (1963).

99. Lorden, G. Procedures for Reacting to a Change in Distribution. *The Annals of Mathematical Statistics* **42,** 1897–1908 (Dec. 1971).

100. Page, E. S. Continuous Inspection Schemes. *Biometrika* **41,** 100–115 (1954).

101. Banerjee, T. & Veeravalli, V. V. Data-Efficient Quickest Change Detection with On-Off Observation Control. *ArXiv e-prints.* arXiv: `1105.1361 [math.ST]` (May 2011).

102. Frank, M., Biedert, R., Ma, E., Martinovic, I. & Song, D. Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. *IEEE Transactions on Information Forensics and Security* **8,** 136–148 (Jan. 2013).

103. McCool, C. *et al. Bi-Modal Person Recognition on a Mobile Phone: Using Mobile Phone Data* in *IEEE International Conference on Multimedia and Expo Workshops* (July 2012), 635–640.

104. Fathy, M. E., Patel, V. M. & Chellappa, R. *Face-based active authentication on mobile devices* in *IEEE International Conference on Acoustics, Speech and Signal Processing* (2015).

105. Y. Sui, a. Z., E.Y.Du & F.Li. *Secure and privacy-preserving biometrics based active authentication* in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2012), 1291–1296.

106. Crouse, D., Han, H., Chandra, D., Barbello, B. & Jain, A. K. *Continuous Authentication of Mobile User: Fusion of Face Image and Inertial Measurement Unit Data* in *International Conference on Biometrics* (2015).

107. Dalal, N. & Triggs, B. *Histograms of oriented gradients for human detection* in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* **1** (2005), 886–893.

108. Samangouei, P., Patel, V. M. & Chellappa, R. *Attribute-based continuous user authentication on mobile devices* in *IEEE International Conference on Biometrics: Theory, Applications and Systems* (2015).

109. McMahan, B., Moore, E., Ramage, D., Hampson, S. & y Arcas, B. A. *Communication-Efficient Learning of Deep Networks from Decentralized Data* in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA* (eds Singh, A. & Zhu, X. () **54** (PMLR, 2017), 1273–1282.

Curriculum Vitae **Pramuditha Perera** **March, 2020**

4 E 30 Street APT 106
Baltimore, Maryland 21218 USA
(+1) 732.208.8369
pperera3@jhu.edu

EDUCATION AND DEGREES

2018–Present Ph.D. Student, Electrical and Computer Engineering
Johns Hopkins University

2015–2018 Master of Science, Electrical and Computer Engineering
Rutgers University, New Brunswick, NJ

2010–2014 Bachelor of Science, Electrical and Electronic Engineering
University of Peradeniya, Sri Lanka

TECHNICAL SKILLS

Programming Skills Proficient in computing tools including Matlab, C/C++, Java, Python, Caffe, PyTorch and MXNet
Theoretical Knowledge Digital Signal Processing, Detection and Estimation, Pattern Recognition, Machine Learning (including deep learning), Data Structures, Computer Vision, Biometrics and Image Processing

JOURNAL PUBLICATIONS

P.Perera and V.M.Patel, Learning deep features for one-class classification, *IEEE Transactions on Image Processing (TIP)*, NOV, 2019.

P.Perera and V.M.Patel, Multiple User Active Authentication on Mobile Devices, *IEEE Transactions on Information Forensics (TIFS)*, OCT 2018.

P.Perera and V.M.Patel, Efficient and Low Latency Detection of Intruders in Mobile Active Authentication, *IEEE Transactions on Information Forensics (TIFS)*, Dec 2017.

P.Perera , S.P.B. Herath, W.S.K.Fernando, M.P.B. Ekanayake, R.I.Godaliyadda, J.V.Wijayakulasooriya. Online Tracking and Event Clustering for Vision Systems, *Journal of the National Science Foundation, Sri Lanka*, Dec 2016.

CONFERENCE PUBLICATIONS

P.Perera ,Vlad Morariu, Rajiv Jain, Varun Manjunatha, Curtis Wigington, Vicente Ordonez, V.M.Patel, Generative-discriminative Feature Representations for Open-set Recognition, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020 (Accepted).

P.Perera , R.Nallapati, B.Xiang, OCGAN: One-class Novelty Detection Using GANs with Constrained Latent Representations, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019.

P.Perera and V.M.Patel, Deep Transfer Learning for Multiple Class Novelty Detection, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019.

P.Perera and V.M.Patel, Dual-Minimax Probability Machines for One-class Mobile Active Authentication, *IEEE Conference on Biometrics: Theory, Applications, and Systems (BTAS)*, Califonia, CA, USA, 2018.

P.Perera , M.Abavisani and V.M.Patel, In2I : Unsupervised Multi-Image-to-Image Translation Using Generative Adversarial Networks, *IAPR International Conference in Pattern Recognition (ICPR)*, China, 2018. **(Best student paper).**

P.Perera and V.M.Patel, Towards Multiple User Active Authentication in Mobile Devices, *IEEE Conference on Automatic Face and Gesture Recognition (FG)*, Washington, DC, USA, 2017.

P.Perera and V.M.Patel, Extreme Value Analysis for Mobile Active User Authentication, from *IEEE Conference on Automatic Face and Gesture Recognition (FG)*, Washington, DC, USA, 2017.

P.Perera and V.M.Patel, Quickest Intrusion Detection in Mobile Active User Authentication *IEEE Conference on Biometrics: Theory, Applications, and Systems (BTAS)*, Niagara, NY, USA, 2016.

PROFESSIONAL EXPERIENCE

Graduate Assistant, Johns Hopkins University, MD, USA. (2018 September - Present)
Research Scientist Intern, Adobe Document Intelligence Lab, MD, USA. (2019 May - Aug)
Research Scientist Intern, Amazon AWS, NY, USA. (2018 May - 2018 September)
Research Scientist Intern, Amazon AWS, WA, USA. (2017 June - 2017 September)
Graduate Assistant, Rutgers University, NJ, USA. (2016 January - 2018 May)
Teaching Assistant, Rutgers University, NJ, USA. (2015 August - 2016 January)
Assistant Lecturer, University of Peradeniya, Sri Lanka. (2015 March - 2015 August)
Teaching Assistant, University of Peradeniya, Sri Lanka. (2014 October - 2015 March)

AWARDS

Awarded best student paper at ICPR 2018.
Awarded Graduate Assistantship of Johns Hopkins University, MD, USA.
Awarded Graduate Assistantship of Rutgers University, NJ, USA.
IESL Best Undergraduate Project in Electrical Engineering in Sri Lanka, 2015.
Winner of IET Present Around the World competition, Sri Lanka, 2015.