Florida Institute of Technology

Scholarship Repository @ Florida Tech

Theses and Dissertations

5-2024

Space Transformation for Open Set Recognition

Atefeh Mahdavi
Florida Institute of Technology, amahdavi@fit.edu

Follow this and additional works at: https://repository.fit.edu/etd

Part of the Computer Sciences Commons

Recommended Citation

Mahdavi, Atefeh, "Space Transformation for Open Set Recognition" (2024). *Theses and Dissertations*. 1431.
https://repository.fit.edu/etd/1431

This Dissertation is brought to you for free and open access by Scholarship Repository @ Florida Tech. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholarship Repository @ Florida Tech. For more information, please contact kheifner@fit.edu.

Space Transformation for Open Set Recognition

by

Atefeh Mahdavi

A dissertation
submitted to the College of Engineering and Science
at Florida Institute of Technology
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Computer Science

Melbourne, Florida
May, 2024

© Copyright 2024 Atefeh Mahdavi

All Rights Reserved

The author grants permission to make single copies.

We the undersigned committee
hereby approve the attached dissertation

Space Transformation for Open Set Recognition by Atefeh Mahdavi

Marco M. Carvalho, Ph.D.
Professor
Electrical Engineering and Computer Science
Major Advisor

Hector M. Gutierrez, Ph.D., P.E.
Professor
Mechanical and Civil Engineering

Thomas Eskridge, Ph.D.
Associate Professor
Electrical Engineering and Computer Science

Munevver Subasi, Ph.D.
Associate Professor
Mathematics and Systems Engineering

Brian A. Lail, Ph.D.
Professor and Department Head
Electrical Engineering and Computer Science

Abstract

Title:

Space Transformation for Open Set Recognition

Author:

Atefeh Mahdavi

Major Advisor:

Marco M. Carvalho, Ph.D.

Open Set Recognition (OSR) is about dealing with unknown situations that were not learned by the models during training. In OSR, only a limited number of known classes are available at the time of training the model and the possibility of unknown classes never seen at training time emerges in the test environment. In such a setting, the unknown classes and their risk should be considered in the algorithm. Such systems require not only to identify and discriminate instances that belong to the source domain (i.e., the seen known classes contained in the training dataset) but also to reject unknown classes in the target domain (classes used in the testing phase). Until recently, the success of almost all machine-learning-based systems has been obtained by conducting them on "closed-set"classification tasks. In such systems, the source and target domains are assumed to contain the same object classes and the system is only tested on known classes that have been seen during training. Different from the "closed set"setting, a more realistic scenario is solving real-world problems consisting of an “open set”of objects. In this dissertation, we propose, develop, and demonstrate an efficient algorithm to improve classification in Open Set Recognition tasks. The proposed technique will explore a new representation of feature space. The efficacy and efficiency of many applications can be improved by integrating OSR, which offers more precise and insightful predictions of outcomes. We demonstrate the performance of the proposed method on three established datasets. The results indicate that the proposed model outperforms the baseline methods in accuracy and F1-score.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my deepest appreciation to my supervisor, Professor Marco Carvalho, for his unwavering support throughout my PhD journey. He consistently conveyed a spirit of adventure in research. Additionally, I am thankful to my committee members for their generous sharing of knowledge and valuable time.

Dedication

First and foremost, I would like to dedicate this dissertation to my beloved husband, Reza, who stood patiently beside me throughout these years. I would also like to express my deepest appreciation to my mom and dad for their support and prayers.

Chapter 1

Introduction

With the advent of building intelligent systems and utilizing machine-learning-based systems, a wide range of applications require robust Artificial intelligence (AI) methods. AI evolves decision-making and alters its dynamics by leveraging intelligent automation that can replicate human mental processes. At present, machine learning systems are widely used in numerous commercial and industrial products, such as autonomous vehicles, video surveillance systems, manufacturing, medical imaging, and so on. These applications often involve the emergence of samples from classes that were not seen during training, commonly referred to as "unknown unknowns"[161].

Handling the "unknown unknowns"can be considered as one of the approaches that enable the system to act robustly in the face of limitations and unmodeled aspects of the world. Traditional training methods operate under the closed-set assumption that all classes encountered during testing are known. However, this assumption becomes problematic when the model encounters an unknown class, leading to decreased performance as it attempts to classify it among known classes (see Figure 1.1).

Ignoring unknown objects causes improper development of the systems and limits their usability. This limitation restricts the application of machine learning systems to

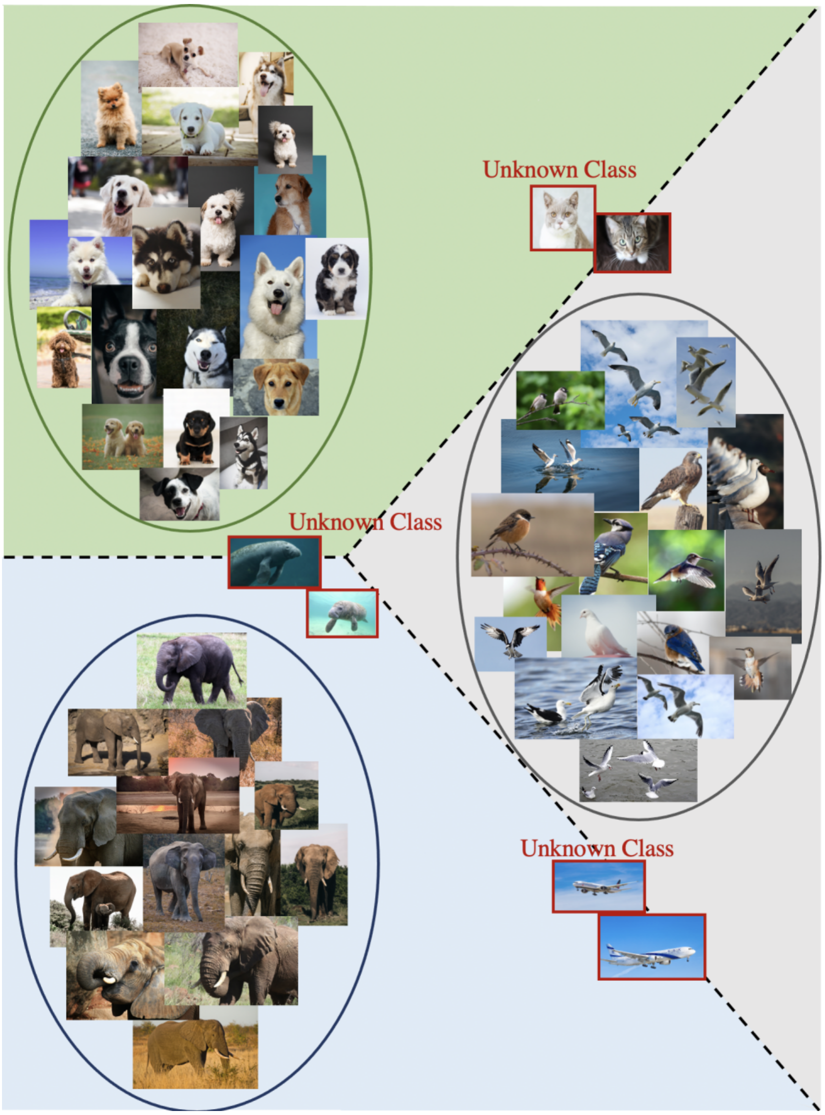


Figure 1.1: A closed-set classifier creates a boundary to differentiate between known classes like dogs, birds, and elephants. However, when faced with unknown samples like cats, airplanes, and manatees during testing, the closed-set classifier categorizes them as known samples with strong confidence. This is because the classifier is not trained to handle unknown samples. Consequently, using a multi-class decision boundary to identify unknown samples is not an effective approach.

known objects and hinders their functionality in real-world scenarios. Just like humans have the capacity to adapt, learn, and make decisions when faced with unfamiliar situations that go beyond their existing knowledge, the dynamic process of decision-making transformation enabled by intelligent automation also necessitates the ability to handle unknown elements. However, constructing a comprehensive and effective model in a dynamic environment poses challenges, as it is impractical to collect, label, and train the model on every possible instance of an unknown object. OSR task involves two objectives: classifying known classes and rejecting unknown classes ([110]). By integrating these goals, OSR enables the development of a more robust system compared to traditional classifiers. This system establishes a more realistic environment and brings benefits to a range of applications such as self-driving cars ([29]), robotics ([176]), e-commerce product classification ([198]), video surveillance ([81]), classification and malware detection ([100]), facial recognition and identifying disruptive images on social media ([124], [84], [91]).

One significant benefit of such a system is improved accuracy and reliability. OSR reduces false negatives that can have serious consequences, like mislabeling a dangerous situation as safe. For instance, a medical diagnostic system might miss detecting new medical conditions in imaging data, leading to false negatives and untreated conditions. OSR ensures that novel abnormalities are correctly identified, reducing the risk of mislabeling dangerous situations as safe. Similarly, in cybersecurity, a lack of OSR could result in the failure to detect new cyber threats, leading to false negatives and critical security breaches. OSR helps the system adapt to emerging threats, reducing the risk of overlooking suspicious activity and strengthening overall cybersecurity defenses.

In the context of credit card fraud detection, businesses face the challenge of identifying new types of fraudulent transactions that may emerge over time. This scenario aligns more closely with open set recognition rather than concept drift, which typically involves changes in the statistical properties of known classes over time. In this context, concept drift could manifest as changes in the patterns or characteristics of known types of fraudulent transactions. However, credit card fraud detection involves the appearance of entirely new types of fraudulent transactions, which is more indicative of open set recognition rather than concept drift. Without OSR, the model may mistakenly categorize these new transactions as non-fraudulent, resulting in losses for the business. By implementing OSR, the model can identify and flag potentially fraudulent transactions, even if they don't match the known fraud patterns in the training data. These transactions are then assigned to an "unknown"or "novel"class for further investigation.

In safety-critical applications like autonomous driving or medical diagnosis systems, OSR can aid in identifying anomalies or outliers in the data, which can offer insightful information for making decisions. For instance, consider a model trained to identify medical images. When encountering an unknown image that it cannot accurately classify, this could indicate the presence of an uncommon or atypical pathology that requires human intervention for diagnosis confirmation and further examination.

Last but not least, OSR improves transparency and explainability. It enables the creation of models that are more transparent and easier to understand. By distinguishing between known and unknown classes, OSR provides transparency by indicating when the model encounters data that it hasn't been trained on. This transparency helps users understand the limitations of the model and the potential risks associated with unseen data. These models can uncover previously unknown patterns in the data, providing more nuanced and comprehensible predictions. For example, in aviation incident reports, OSR enhances incident categorization, risk assessment, and decision support [18]. It also has the ability to identify risks and potential safety hazards that were previously unrecognized [70]. These capabilities provide decision-makers with a broader perspective, allowing them to gain a more accurate understanding of the organization, identify areas for improvement, and seize new opportunities. In the past literature, authors have proposed terms such as "open set recognition"[12], "open category learning"and "open-world recognition"[11] that can respond to model failure. In this work, we will use the term open set recognition.

Although open set recognition operates within its distinct domain, there are connections to domain adaptation [132, 35], transfer learning [127, 193, 181] and few/zero-shot learning [191, 174, 54, 49]. Domain adaptation aims to address the discrepancy between the distribution of the training data and the distribution encountered during deployment. Transfer learning involves leveraging knowledge from a well-labeled source domain to enhance learning in a target domain where labeled data is limited. Few-shot learning deals with scenarios where the model is trained with only a few examples per class. By understanding and leveraging insights from these fields, open set recognition techniques can advance towards more robust and adaptable models in real-world applications.

There are two broad categories of OSR systems. The first one refers to the task of discriminating known class instances from unknown class instances. This mechanism which is not able to distinguish between the known classes acts as a detector rather than a classifier. This technique is applied in research such as [16, 161]. In the second category, in which the number of classes is more than two, OSR is concerned with distinguishing between the known classes. This system identifies unknowns and labels the input as one of the known classes it best fits or as unknown [58, 83, 11, 12]. A challenge faced by a potential solution to the OSR is estimating the correct probability of all known classes and maintaining the performance on them, along with a simultaneous precise prediction of unknown classes and optimizing the model for them.

1.1 Problem Statement

Let K be the total number of distinct known classes, and $D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ be a set of training dataset with N samples $x_i \in X = \{x_1, x_2, ..., x_N\}$ and their corresponding labels $y_i \in Y = \{C_1, C_2, ..., C_K\}$. Open set recognition involves learning a function f that can correctly classify an instance to one of the known classes, or an unknown class. For an unseen instance $x^{'}$ (not in X), if $y^{'} = f(x^{'})$ is the class label that f predicts, $y^{'}$ might be either one of the recognized K classes in the scenario of a closed set or a new class in the event of an open set. In contrast to closed set classification, the problem of open set identification involves handling unknown examples that may correspond to classes that were not known during training. In other words, the learner is robust and can effectively manage unknown instances. The fundamental obstacle to open set recognition is this distinction.

More precisely, in the closed set classification task, the learner only has access to a fixed set of known classes $Y = \{C_1, C_2, ...C_K\}$ and constructs a classifier during the training phase. The resulting classifier is tested on the data from only the K classes. However, a problem emerges with the appearance of a test sample from an unknown class which does not belong to any of the known classes. Thus, the most likely class for an input observation is always provided and an unknown will wrongly be recognized as a sample belonging to one of those pre-defined classes. In OSR, however, knowledge of the entire set of possible classes cannot be considered during training or any other time. The classifier is allowed to predict classes from the set of $Y^{'} = \{C_1, C_2, ..., C_K, C_{K+1}, ..., C_{K+\Omega}\}$, where classes C_{K+1} through $C_{K+\Omega}$ cover all unknown classes not observed during training but which appeared at query time. A test sample may be predicted to belong either to one of the known classes $C_i \in Y$ or to an unknown one.

The difference between OSR and traditional classification is visualized in Figure 1.2. The decision boundaries in Figure 1.2(a) are create by training a traditional Nearest Class Mean (NCM) classifier on three different known classes illustrated by diamonds, circles, and squares and the unknown inputs represented by stars. Figure 1.2(b) demonstrates the distribution of original dataset in the open space when zooming out from the closed three-class model. Having incomplete knowledge of the entire set of possible classes, this classifier assigns class labels from the closed training set to an unlimited region. Therefore, at the classification time, the unknown inputs in the open space will be misclassified. On the other hand, OSR discriminates known samples and limits the scope of decisions by the support of the training data (see Figure 1.2(c)). Hence, OSR necessitates specialized techniques because its objective is confined and strict decision boundaries.

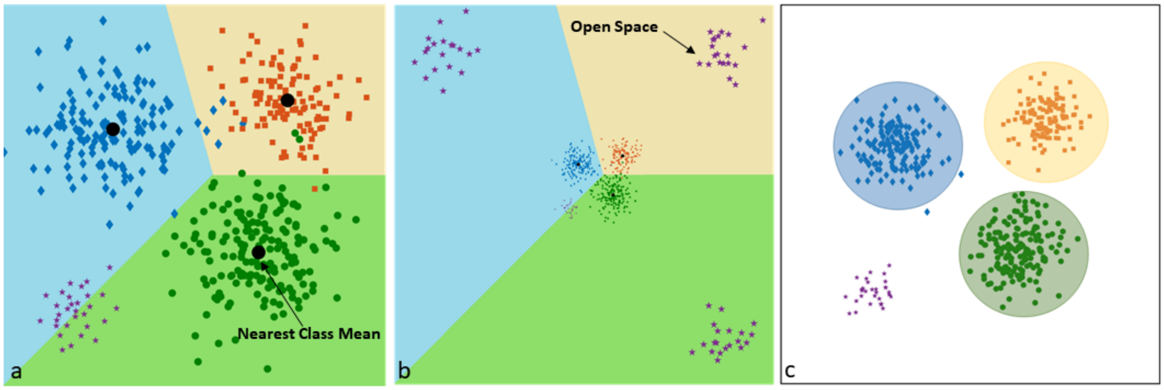


Figure 1.2: An overview of the issue of OSR. (a) A closed set classifier is incapable of detecting unknown samples since it can only learn decision boundaries that divide the feature space into three sections. (b) Zooming out from the closed three-class model. (c) Open set recognition which is preferred to have strict decision limits around the known classes. Consequently, unknown space is defined as any area outside of boundaries.

1.2 Approach

In this dissertation, we focus on representation learning, also known as feature learning, for OSR. Representation learning refers to the process of acquiring data representations that facilitate the extraction of meaningful information. It enables a machine to automatically find the representations required for detection or classification after being fed with raw data. The other options for open set recognition include methods that involve borrowing or generating additional data. In borrowing additional data, techniques utilize unknown examples to expand the training set. Unknown data is introduced during training to better discriminate between known and unknown classes, thereby improving feature learning. However, it's important to note that in these approaches, the unknowns introduced are not the actual open set; rather, they are known unknowns. Known unknowns represent entities that are expected to not be classified within the known classes. On the other hand, techniques that do not involve additional data often employ a confidence threshold set on the softmax probabilities. This threshold helps differentiate between samples confidently classified as known classes and those with lower confidence scores, which are likely unknown. Samples falling below the threshold can then be classified as unknown. The proposed technique here falls into the category of methods that do not require additional data.

Deep Neural Networks (DNNs) are used to minimize error, and during this minimization process, a set of representative features are provided that can be driven by various goals. A typical classification neural network consists of an input layer, hidden layers, and a classification layer. By employing loss functions such as cross-entropy, the hidden layer representations are trained to reduce classification loss of the output in closed-set classification scenarios. However, it is important to note that the representations may not inherently contain the desired feature space that proves useful for

OSR.

We propose a mechanism to enhance a neural network's feature space representation for better detection of unknown situations and handling OSR. This strategy is built on expanding the existing loss functions with a new type of loss that we refer to as "Superlative Loss". In the proposed algorithm the between-class separation is maximized in terms of the distance between class means of all the known classes. Then, during neural network training, our objective is to create a desired feature space where classes are well-separated, which is advantageous for open-set recognition. This is achieved by distorting the original feature space learned in the closed-set classification task to enhance open-set recognition capabilities.

As the hidden layers of DNNs can be viewed as multiple levels of input representations, the superlative loss procedure revolves around determining how to replace the original feature representation with one that is more advantageous for OSR. This new representation can be learned in such a way that different classes are further apart and well separated which leads to larger spaces among them. Consequently, unknown examples can easily be detected. Our proposed idea builds upon the concept of inter-class separation and intra-class compactness, initially introduced in the triplet loss function proposed by [168]. The triplet loss function aims to reduce the distance between similar items, such as images of the same person, while increasing the distance between dissimilar items, like images of different people. Numerous studies have explored methods to maximize inter-class separation and minimize intra-class compactness [168, 24, 131, 199, 201, 108, 72, 130]. In our proposed approach, we aim to achieve this objective by introducing an optimal boundary that encompasses all classes through Principal Component Analysis (PCA). PCA is utilized to define this boundary in the feature space, facilitating optimal separation between classes. During optimization, we minimize "Superlative Loss"function, which iteratively adjusts the positions of classes, moving them towards the boundary to maximize inter-class separation while minimizing intra-class compatcness. This iterative process ensures that the feature space adequately accommodates sufficient separation between classes.

Suppose K represent the total number of distinct known classes, and let $D = (x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$ denote a training dataset consisting of N samples $x_i \in X = x_1, x_2, ..., x_N$ and their corresponding labels $y_i \in Y = 1, 2, ..., K$. OSR involves learning a function f that can accurately classify an instance into one of the known classes or an unknown class. For an unseen instance $x^{'}$ (not present in X), if $y^{'} = f(x^{'})$ represents the class label predicted by f, $y^{'}$ can potentially belong to one of the recognized K classes in a closed set scenario, or it can indicate a new class in an open set scenario.

1.3 Contributions

Our contributions include the following:

- The proposed method is effective for OSR problems in DNNs and is flexible to be used with different types of loss functions on any neural architecture.

- We introduce superlative loss for developing an ideal feature space representation that makes OSR easier without borrowing additional data or generating them. Thus, this robust model does not require complex network architectures which can be costly and time-consuming.

- We demonstrate that superlative loss delivers statistically significant improvements in terms of overall F1 score and accuracy when applied to two different types of loss functions on three datasets.

1.4 Dissertation Structure

This dissertation is structured as follows. Chapter 2 introduces an extensive survey on open set recognition, as outlined in the paper 'A Survey on Open Set Recognition' [110]. This paper was published in the proceedings of the 2021 Artificial Intelligence and Knowledge Engineering (AIKE) conference. In this chapter, we explore various open set recognition approaches and provide a comparative analysis of their respective advantages and disadvantages. Chapter 3 and Chapter 4 present the proposed methodology and the experimental evaluations and comparisons. These findings are derived from the paper 'Informed Decision-Making through Advancements in Open Set Recognition and Unknown Sample Detection' [111], which was published in the proceedings of the 57th Hawaii International Conference on System Sciences in 2023. Finally, Chapter 5 concludes the dissertation.

Chapter 2

Literature Review

In this chapter we summarize the various research efforts for Open Set Recognition. We briefly differentiate OSR from multi-class classification problem and discuss their limitations in section 2.1.1. In section 2.1.2 applications of OSR are highlighted. Following that, we propose the existing works about OSR and distinguish their respective advantages and disadvantages in section 2.2.

2.1 Background

2.1.1 OSR vs. Multi-class Classification

OSR is referred to as a classification-based task. Most of the approaches to OSR were formed based on regular classifiers due to their closeness to the classification task; however, the adaptation of a classifier which is valid for OSR is not always possible. Classification and anomaly detection are the closest relatives to OSR. The relationship between OSR and these related areas is summarized in Table 2.1. There are many approaches regarding classification with a reject option in the literature

Table 2.1: Relationship between OSR and the related areas

Settings	Training Data	Testing Data	Tasks
`Traditional Classification`	Known	Known	Classifying known data
`Anomaly/Outlier Detection`	Known	Known/Unknown	Identifying rare items
`Open Set Recognition`	Known	Known/Unknown	Classifying known data and rejecting unknowns

[8, 75, 53, 204, 56, 59, 192, 55, 65, 209] which have been adjusted to support open sets. In a threshold-based classification strategy, an instance is recognized as unknown if the matching score to the most likely class is below the established threshold, i.e., the sample is far away from all training samples [42, 119]. For instance, Phillips et al. [135] described an evaluation protocol for open-set face recognition algorithms which decides whether the identity of a sample corresponds to a known class or not if the similarity score exceeds an ad-hoc rejection threshold, and then reports the identity of the accepted sample. Another work [188] estimated a rejection threshold based on the ratio of the two highest decision scores obtained from a vote list ranking. This method combines hashing functions and classification methods. Whenever a face query is requested, it is compared to all hashing functions and the vote list is generated based on their response values. A transduction-based study which considered open-set face recognition from an evaluation point of view is introduced in [99]. The proposed open set algorithm uses distances of a test image to its k-nearest neighbors in both inter and intraclass and extracts the credibility values' distribution. This method rejects a face as unknown if the highest credibility value passes a proper threshold. However, defining such a threshold for an unknown is the critical part of all the approaches adopting the threshold-based classification scheme. For instance, in a task of image classification, when an image is slightly different from what the network learned, adding a threshold to the classification output may reject the image as unknown. So, thresholding depends on the operating environment of a recognition system and how distinct the class is.

Moreover, this technique does not work very well as all outliers of each class may be classified as unknown and rejected. The other challenge with thresholding is detecting adversarial images trying to bypass machine learning systems to misclassify.

The greatest part of rejection-adapted approaches rested upon variants of Support Vector Machine (SVM) [32] classifiers with the ability to reject observations [55, 209, 65], and the one-class classifiers based on support vectors [28, 71, 166, 79, 185]. Although such techniques are related to OSR in the sense of rejecting an input, they have different reasons to do rejection actions. Classifiers with rejection options focus on the ambiguity between classes to reject an uncertain input of one class as a member of another one and minimize the distribution mismatch between the training and testing domains, while OSR rejects an input because of not belonging to any of the known classes. Chow [31] derived optimal thresholds to optimize the ambiguous regions between classes in multiclass classification task with the assumption of known prior probabilities of classes. Therefore, rejecting uncertain inputs in such classifiers protects misclassification but is not enough to handle unknowns. They have infinite positively labeled open space and infinitely open space risk and thus are not able to solve OSR problems formally. In these techniques, unknowns often appear to be uncertain and are labeled with confidence. In contrast, OSR supports rejecting the unknown object by discovering the acceptable amount of uncertainty and searching among any of the known classes to identify if the true class exists. In other words, the set of possible outcomes of predictions is an important difference between OSR and a typical multi-class classifier.

For example, in margin classifiers like SVMs, confidence is evaluated in terms of an associated distance to the decision boundary given for each example. The goal of SVMs is to find an optimal hyperplane to classify and separate the classes of training samples. The hyperplane defines half-spaces and divides examples of the separate categories by maximizing the distance between itself and the nearest training points. In such classifiers uncertainty is high near the decision boundary and confidence will be increased with distance from the decision boundary; the farther an input is from the margin, the more confident one can be that it belongs to the known classes. Thus, an unknown far from the boundary is incorrectly labeled and will be incorrectly classified with very strong evidence. For example, in Figure 2.1(a), a plane found by the SVM separates bicycles and airplanes and maximizes the SVM margin making "airplane" a half-space. An unknown ("?") far from the training data will be misclassified and likely be labeled "airplane" as the label propagation is not limited.

For such classifiers that use observation-to-margin distance as the only information to identify unknowns, resting on a threshold as a confidence rate for rejection is not enough for discovering the hidden unknown classes. Moreover, due to incomplete information about unknown classes, selection of the decision threshold depends merely on the knowledge of known classes, and the decision score calibration is processed implicitly by closed set assumptions. Therefore, OSR cannot use rejection-adapted SVM as a good option, although it outperforms a multi-class SVM which strictly assigns a label to the known. Additionally, there exist limitations in probabilistic models for the open set problem where the prior probability of the classes is unknown and Bayes' theorem is violated. Considering the likelihood of unknown classes, Bayes' rule cannot be exactly utilized as Bayesian posterior probability. This model, which holds closed world assumptions, cannot be modeled for unknown classes unless the probability of all unknown classes is assumed as known. On the other hand, an obvious approach to add a rejection option to a multi-class classifier is to incorporate a thresholded probability model, in which a decision threshold is added into a posterior probability estimator, $P(L \mid X)$ [95, 80]. Where $L \in N$ is a particularly known class label for a fixed set of N known classes and X is an input sample. At the time of the appearance of unknown classes, a given data is labeled as unknown if the maximum probability over known classes is below the defined threshold. However, there is a chance that a misclassification still exists due to unlimited open space risk.

On the other hand, some people argue that identifying novel classes [14, 1, 134], discovering outliers [150, 195, 202] and detecting anomalies [25, 61, 158] sometimes can solve the OSR problem [15, 97, 169]. Although these methods such as one-class support vector machine (OCSVM) [166] or support vector data description (SVDD) [185] referred to the problem of identifying unknown data and have been a good start for OSR, the problem setting is different from that of OSR. These techniques are restricted to merely solve OSR for One-class classification problems [184, 90] in the one-class setting. One-class classification is solved by finding a decision function f which transforms input data into a high dimensional feature space. The function f is positive in some small region corresponding to one class of objects, and negative elsewhere. This algorithm tries to find a separating hyperplane which maximizes the margin between the training data (positive examples) and the origin (considered as negative examples) (see Figure 2.1(b)). An one-class classifier then is trained to label a test example x as an outlier if $f(x) < 0$, and normal if $f(x) > 0$.

Although the possibility of modeling each single class [186] or concentrating multiple known classes into a single one opens up a new way for multi-class novelty recognition [16, 183], these techniques alone are not sufficient for creating a balance between the risks of the unknown and multiclass recognition for OSR, leading to poor performance. Compared to some anomaly detection techniques where an auxiliary dataset of outliers is accessible at the training time [73], OSR problems do not have access to unknown classes. A number of surveys have been written to analyze and discuss the concept of outliers/anomalies from different points of view [114, 112, 113, 136, 26, 211, 3, 77]. These techniques with a very long history in machine learning can model normal data,

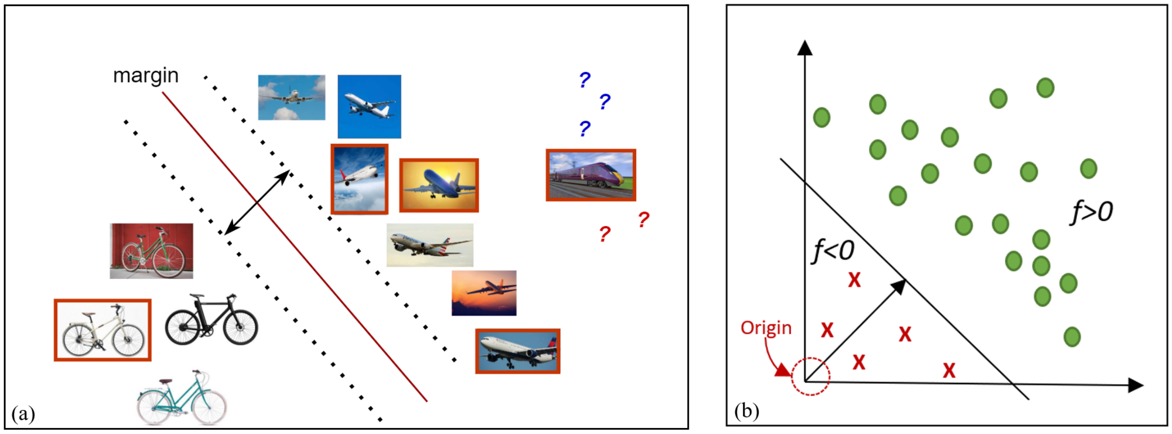


Figure 2.1: (a) Two-class SVM classifier. Images with an orange box are from testing and the rest of images are from training. Testing images can be known ("bicycle", "airplane") or unknown ("train," "?"). (b) One-class SVM classifier.

then find a distance from the class mean for each sample and place an optimal threshold for discovering abnormalities. In the case of the existence of an appropriate threshold on one or more one-class classifiers [99, 186, 106], a finite open space risk will be produced and OSR can be supported. However, these techniques must have a robust performance that requires trading off between maximizing the recognition rate and minimizing the inclusion of novel data. Moreover, they achieve less stability and worse performance over OSR models once classes are withheld during training.

2.1.2 Applications of Open Set Recognition

In this section, we will make a general review of some practical applications of OSR. Emerging real-world recognition systems require OSR to recognize unknown inputs and learn them when needed. There is a multitude of real-world application domains where OSR can play a role, such as cyber-physical systems, intrusion recognition, face identification, video tracking and surveillance, image and text classification, spam filtering, forensics linguistics, movie genre classification, and document tagging [9, 94, 164, 153, 133, 122, 33, 151, 120, 173, 167, 139, 7, 96, 27, 196, 197]. OSR is a challenging task in a large number of safety environments where even a small fraction of errors on unknowns could place human lives at risk, such as a self-driving car defect or robotic surgical assistants with flaws in perception and execution [206, 179, 145]. Moreover, real-world robots can expand their knowledge if they will be able to detect unknown objects, discover the need to learn about them and learn them continuously

An automatic face recognition system that is usually encountered with unknown individuals [30, 137, 69, 6, 86, 118, 68, 82] is another domain to be deployed in open-universe scenarios. There is a wide range of real applications of face recognition, for instance, reducing retail crime, controlling mobile phone access, helping police officers, identifying people on social media platforms, and so on. This system consists of a feature extractor and a match component to do the face recognition. After feeding a face image into the system and extracting the biometric information, a match component compares the extracted features with the stored gallery faces. Face matching includes two different tasks: face verification and face identification. The face verification problem is to compare a pair of face pictures to decide whether the two face images represent the same individual or not. In the face identification, the comparison is against a gallery which contains a set of face images to recognize the corresponding identity of a given face picture. Although the face identification problem finds the nearest identity to the querying face, it can be treated as an open-set problem, and thus we need to decide whether the query subject is registered in the gallery or not. Initially, OSR is introduced by Li and Wechsler. [99] for face recognition task. There are a few works [13, 175, 43, 103, 177] which are mainly based on incorporating an operating threshold on similarity scores to address this problem. Other research [194, 124, 10] points to the problem of face recognition with real-world databases in social media posts to determine and associate the most probable identity for the query face sample automatically.

Another domain where OSR can be a solution is malware classification for cybersecurity. Malware, shorthand for malicious software, meets the harmful intent of cyber-attackers which is designed to pose severe and evolving security threats to individuals, government organizations, and private institutions. In the Internet age, with a higher frequency of communications among computer applications and their respective refinements, the number of new malware samples has explosively increased. Hence, it is required to keep up with the sophistication of newly received attacks and develop intelligent methods for effective and efficient malware detection. This domain is faced with the challenge of incomplete knowledge of the training data because of emerging novel types of malware. The ever-changing nature of malware, as the intruders are continuously altering network attacks to bypass the existing detection solutions, calls for the development of autonomous countermeasures and the recognition of novel malware classes [74, 34]. Rudd et al. surveyed many existing intrusion detection algorithms and proposed an open-world mathematical framework to extend and obviate the closed world assumption behind them [157]. This flawed assumption impedes mappings between a machine learning solution and realistic malware recognition problems in which knowing all types of possible attacks cannot be known a priori.

Activity recognition has practical applications to facilitate human-vehicle communication and the transition to the level of driving automated systems. This task has the potential to recognize driver distraction for safety and improve dynamic driving adaptation like turning on the light if the person is reading a book or adjusting the seat while drinking coffee. However, it is difficult to apply computer vision models inside the vehicle cabin because of the dynamic nature of the surrounding environment. We cannot capture all possible driver behaviors in the training data, then the model, developed for closed set recognition, will be quickly exposed to uncertain situations and put the driver in disturbing and potentially dangerous situations. In [152], the task of open set driver activity recognition is introduced to address this issue.

2.2 A Categorization of OSR Techniques

2.2.1 Overview

Scheirer et al. [161] introduced the first formalization of OSR by balancing open space risk R_O associated with labeling data that is far from known training samples against minimizing empirical risk R_ϵ over training data. By assuming f to be a measurable recognition function, open space risk $R_O(f)$ for known class k is described as the following:

$$R_O(f) = \frac{\int_O f_k(x)dx}{\int_{S_k} f_k(x)dx} \tag{2.1}$$

This formalization provides the proportional value of positively labeled open space O against to the total measure of S_k consisting all of the known positive training samples $x \in k$ as well as the positively labeled open space. This paper argued that the essential element of OSR is finding the recognition function f, where $f(x) > 0$ indicates the positive recognition of the class k of interest. This function is defined as a minimization of the open space risk to capture the risk of labeling the unknown samples as known, beyond the sensible recognition of the training data, as follows.

$$argmin_{f \in H} \left\{ R_O(f) + \lambda_r R_\epsilon(f) \right\} \tag{2.2}$$

where λ_r is the regularization tradeoff between open space risk and empirical risk. This study proposed a “1-vs-set Machine” which consists of two parallel hyperplanes. The proposed formulation with a linear kernel balances empirical and open space risk by exploiting a second hyperplane from the marginal distances of a 1-class or binary SVM.

The main hyperplane is a base SVM which defines half-spaces and aims at maximizing the margin. The second hyperplane is added in such a way as to minimize the positive labeled region bounded between two planes and handle open space risk. This method defines a definition that generally describes region of known classes for each individual binary SVM, however, it lacks the procedure of distance measurements. It even does not clarify the space for measurements of such distances. Therefore, it cannot bound the space that each known class belongs to that leads to the existence of open space risk. Inspired by this technique, Cevikalp [23] found the best fitting hyperplanes by placing them close to the samples of one class and far from the other class samples. There have been many more attempts over the past years to address open space risk for training OSR models. Followup works by Scheirer et al. [162, 83] were inspired by the fact that leveraging Extreme Value Theory (EVT) [22] on the SVM decision scores provides better performance than exactly applying the raw score values. Both approaches have proposed EVT-based SVM calibration techniques to enable the SVM-based classification to deal with an open-set setting. Other OSR algorithms such as [12, 201, 156, 160] also include EVT to analyze the association of a data point with an unknown class. Extreme value modeling has been increasingly used to analyze post-processing scores and enhance the performance of OSR. This theory is meant to study a level of confidence by determining the fraction of objects deviating from the expected value. EVT is effectively used in many research areas such as environmental risk management, finance, insurance, anomaly detection, or network monitoring.

The following is the definition of EVT:

Let $\{X_1, X_2, X_3, ..., X_n\}$ be a sequence of independent random variables with unknown distribution function $F(x)$, and $X_m = \underset{i}{max} X_i$, $i \in [1, n]$. Assume there exists a pair of sequences (a_n, b_n) with $a_n > 0$ and $b_n \in R$ such that $\lim_{n\to\infty} P\left(\frac{X_m - b_n}{a_n}\right) = G\left(x\right)$. Then if G is a non-degenerate distribution function, it belongs to one of Fréchet, Weibull, or Gumbel distribution families. These three distributions can be combined into a single general form which is called Generalized Extreme Value (GEV) distribution and is defined in Equation 2.3:

$$E\left(x;\mu,\sigma,\xi\right)=\exp\left\{-\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{\frac{-1}{\xi}}\right\} \tag{2.3}$$

Where μ, σ and ξ are the location, scaling, and shape parameters, respectively. These three models provide the data distribution for a reasonable evaluation of the probability of occurrence of rare events. As extreme values appear in the tails of the distributions, EVT examines the distribution tails and aims to predict the probability that a given sample is an extreme value applying Equation 2.3. For OSR, EVT models the probability distributions of the match and non-match recognition scores and the rejection threshold is usually estimated from the overlap region of extreme values found in the tails of probability distributions.

Unlike SVM, which divides all the space with hyperplanes to define sections and allocate them to one of the current classes, Scheirer et al. [162] applied EVT and Weibull distributions to build such hyperplanes without dividing the whole space. This work introduced the idea of a compact abating probability (CAP) model based on a one-class classifier which, when thresholded, can further limit the open space risk. The labeled region is limited and the open space risk is minimized if the value of the probability of class membership is decreasing in all directions as samples pull out of the training data and move towards the open space. Distribution of decision scores for unknown recognition is considered by extending “1-vs-Set Machine” to W-SVM (a Weibull-calibrated non-linear classifier). This algorithm yields better modeling for a binary SVM at the decision boundaries by applying EVT for score calibration and combining One-Class SVMs using a Radial Basis Function (RBF) kernel with the scores from multi-class SVM. The first application of the W-SVM algorithm is to the difficult problem of fingerprint spoof detection where inter-class distances between a live finger and an effective spoof one are small in the feature space [146]. Another work [34] based on W-SVM is proposed for open set intrusion detection on the KDDCUP'99 dataset.

Based on this intuition, Jain et al. [83] introduced a variant of W-SVM that is called Support Vector Machines with Probability of Inclusion (PISVM). This algorithm formulates the multi-class OSR problem as one of the modeling positive training data at the decision boundary. An SVM with RBF kernel is utilized as a binary classifier for each class and trained by the One-vs-All approach, where the samples of the remaining classes are assumed as negative. It models the unnormalized posterior probability of inclusion for multiple classes as a basis to reject unknown samples. Then, it fits probability distributions consistent with the statistical EVT, leveraged on the decision scores from the positive training samples. For a given sample, a class is chosen whose decision value makes the maximum probability of induction. The sample is recognized as unknown if that maximum is under a predefined threshold. Although the proposed algorithm is more accurate than W-SVM, it did not always confine open space risk, the issue that occurred with a regular SVM.

In spite of being a recent research focus, EVT for OSR does not merely guarantee a bounded open space, as PISVM [83] does not always bound open space and W-SVM [162] relies on one-class models to bound open space rather than counting on EVT models. Compared to EVT-based models, a simpler algorithm called Specialized SVM was proposed by Júnioret et al. [88] recently. This algorithm bounds the represented space for known categories and provides a finite risk of the unknown if using an RBF kernel and limiting the bias term to be negative. Additionally, the proposed EVT-based calibration of 1-vs-rest RBF SVMs modeling in both W-SVM and PI-SVM has two deficiencies. The first deficiency is that it is not ideal for OSR which requires incremental updates. Supporting incremental learning is a principal goal in designing an algorithm for OSR, especially one that is constantly used over a long period of time. This approach cannot add novel detected objects and tune the model to enhance the fit as a new item arrives. So it is not able to learn the model incrementally. The second deficiency is that it does not address the fundamental issue of choosing thresholds, which requires prior knowledge in such threshold-based classification models. However, the authors of the last two papers recommended choosing the thresholds according to the problem openness, which is not reasonable since the openness is not usually known in the corresponding problem.

To tackle these deficiencies Scherreik et al. [165] formulated the probabilistic open space SVM (POS-SVM) which rests on a one-vs-all binary SVM. An individual reject threshold for each of the known classes is computed and optimized by a validation set. Platt's method [138], the most widely used probability estimator, is also used to convert SVM scores to a calibrated probability estimation. Another possible approach more appropriate for the open-world with incremental learning capabilities proposed a Nearest Non-Outlier (NNO) algorithm [11]. NNO adapts the Nearest Class Mean Classifier (NCM) [117], the basis of most open-set classifiers, for OSR by using non-negative combinations of abating distance. This work is built on the concept of a CAP model; however, it generalizes the model to gain zero open space risk by applying a threshold on any non-negative combination of abating functions. NCM represents the classes by the mean feature vector of their components, and a test sample is set to a class with the closest mean using Euclidean distance between the class mean and the test feature vectors. The NNO algorithm was inaccurate because of using thresholded distances from the nearest class mean. Additionally, it does not tune the rejection threshold automatically as new classes arrive and the problem evolves. So this algorithm does not properly model the dynamic nature of open-world recognition. To mitigate this problem, Rosa et al. [38] used the Hoeffding bound [78] to incrementally update the threshold for an unknown class, instead of estimating it from an initial set of known classes and keeping it fixed as previously used in [11].

Statistical approaches such as threshold-based decision technique are being widely employed in text document open-set classifications [48, 41]. Probably, cbsSVM [48] is the first open multiclass text classifier. This model is based on the CBS (Center-Based Similarity) space learning method [47], whereby a center for each class in the original problem is computed first. Then the data is transformed into a vector of their similarities to the class centroids to limit positive labeled area from an infinite space to a finite space. A decision threshold is then applied on posterior probabilities which are estimated from the SVM scores for each classifier using Platt's algorithm [138] to identify unknown classes.

Doan et al. [41] represented Nearest Centroid Class (NCC) which is incremental learning and built upon the NCM algorithm. Instead of using the class mean for each class member, this model is based on a series of closest neighbors of the centroid class. In spite of its similarity with NNO in terms of using multiple centroids, the proposed model addresses the issue of the new classes being added incrementally related to NNO and updates information for a class ball. During training, this algorithm attempts to create the boundary region for each known class. Each class is a set of balls centered at class centroids where each ball represents a number of its data points. An observation is treated as an unknown when not any of the nearest class boundaries support it.

Additional works like Assign-and-Transform-Iteratively (ATI) [128], LACU (Learning with Augmented Class with Unlabeled data) framework [37] and Separate to Adapt (STA) [105] require the help of unknown source samples. Additionally, these methods maintain the assumption of containing unknown classes in the source domain. Busto et al. [128] utilized unknown source samples whose class does not overlap with that of the unknown target. This algorithm maps the source domain's feature space to the target domain. It learns this association by minimizing the distance from target samples to each of the source classes' center. Based on a binary linear program, the assignment problem is defined that also implicitly handles outliers by discarding predicted unknown target samples not connected to any of the source domain's samples. This process iterates over the converted source samples to repeat the process of solving the assignment problem, approximating the mapping from one domain to another one, and updating the transformation until it converges. After convergence, linear SVMs are trained in a one-vs-one setting over the converted data to label the target domain. In this work, the execution of a typical SVM is compared with an alternative model introduced in [162]. The other work [37] presented the LACU-SVM approach to address OSR by exploiting an unlabeled dataset besides the training set and tuning the decision boundary. Based on the large margin principle from the SVM algorithm, classes should be divided by large margin separators. Thus, the unlabeled data can identify large margin separators that have similar performance to the seen classes when adopting the one-vs-rest approach. LACU then selects one of these separators that is closest to the labeled region. Distinguishing augmented (unknown) classes involves the utilization of the LACU-SVM in which seen classes are surrounded by large margin separators. Then, it picks a classification boundary among all low-density separators that minimizes the misclassification risks among the seen classes as well as between the augmented and the seen classes simultaneously.

Unlike several methods proposed in the literature to address OSR, Liu et al. [105] recently took into account the openness [161] of the target domain, which is measured by the proportion of unknown classes to be identified in the target domain. In this work, a multi-binary classifier is trained in a one-vs-rest setting to measure the similarity between the entire target domain and each source class. All the target samples are ranked by such similarity. Then, a binary classifier is trained using samples with the highest/lowest similarity to separate all target samples and generate the weights for rejection. These two steps are repeated and samples of unknown classes are rejected progressively in the adversarial domain where one more class is added to the source classifier for the unknown class.

Web genre identification (WGI) is considered as a multi-class text classification task with the ability to automatically recognize the genre of web documents. Therefore, search results can be categorized based on the genres that not only facilitate retrieving information but also provides rich descriptions of documents and enables more specialized queries. Instead of the content, WGI puts the emphasis on the relation of form and style with their associated web pages [115, 154]. In an experimental study on the open-set classification models for WGI setup, Pritsos et al. [143] examined one-class SVMs and Random Feature Subspacing Ensembles (RFSE) [144] models. With respect to this fact that most of the complementary information to differentiate known from unknown samples is placed in the tail of a distribution, modeling the tail of match and non-match error distributions can help to find the optimal threshold for a given recognition model. Inspired by this intuition, Zhang et al. [207] extended the Sparse Representation-based Classification (SRC) algorithm to OSR. This algorithm models the tails of these two residual errors using EVT. The identity of an unknown test sample and open-set identification is determined by getting the confidence score for that sample and hypothesis testing.

Extreme Value Machine (EVM) [156] as a probabilistic framework for open set classification also considers Weibull distributional information when learning recognition functions. EVM is the first classifier to perform a nonlinear RBF approach motivated by EVT and provides a more powerful representation model for OpenMax which will be discussed in section 2.2.2. Using CAP models, EVM is able to bound open space.

[74, 68] are applications of EVM in intrusion detection and open face recognition respectively. However, this approach has drawbacks with regard to the choice of the threshold which controls the open set classification error and more important, strongly relies on the relative arrangement of the known classes. EVM assumes that the behavior of the unknowns can be inferred by the geometry of the known classes, and thus the recognition task may fail when the known and unknown geometries of classes are different. To overcome these limitations, two robust algorithms [190] derived from EVT that do not rely on the geometry of the observed data. These classifiers, called generalized Pareto distribution (GPD) and generalized extreme value (GEV), utilize the intuition that new points to be classified as known or unknown are more likely to be unknown if they are far away from the training data. Moreover, these algorithms are efficient to update upon arising new training data.

There are also a few studies [142, 88, 86] utilizing Nearest Neighbor models on this topic. Júnior et al. [87] proposed the Nearest Neighbor Distance Ratio (NNDR) classifier, which in turn, is a multiclass open-set extension for the Nearest Neighbor (NN) algorithm and is referred to as Open Set NN (OSNN). During the prediction phase, the OSNN first finds the nearest and second nearest neighbors y and z regarding a test sample t in order that $\omega(y) \neq \omega(z)$, where $\omega(s) \in L = \{l_1, l_2, ..., l_n\}$ represents the class of sample s and L is a set of training labels. Then, this classifier calculates the similarity scores' ratio and applies a threshold to recognize sample t as unknown having low similarity. This ratio is defined by $r = d(t, y)/d(t, z)$, where Euclidean distance of two samples s and $s^{'}$ is shown by $d(s, s^{'})$. Recently, Pritsos et al. [142] viewed WGI as an open-set task and applied the NNDR algorithm to its setup to better deal with incomplete genre palettes.

2.2.2 Deep neural network-based algorithms

Following the extensions of traditional classification algorithms for OSR, there is a considerable amount of research in developing deep neural networks for OSR in the literature [12, 58, 19, 20, 201, 170, 171, 125, 39, 155, 189]. However, with the shift to deep networks, which combines learning features and learning the classifier, the performance of the system for OSR is still far from optimal [17]. Researches have addressed this problem by thresholding on the Softmax scores. Moreover, an effective rejection solution than thresholding softmax is using a garbage or background class which has dominated most of the modern detection approaches like [148, 210, 107]. Such background-class-based modeling can tackle the problem of unknowns in neural networks by adding another class as representative of unknown samples during training. Although this approach works well for datasets like PASCAL [44] and MS-COCO [104], it is a probable source of negative dataset bias [187] and has limitations in the real world with infinite negative space of infinitely many unknown inputs to be rejected.

According on the components of the training set, existing OSR procedures are categorized into three groups: training with additional data that was borrowed, training with additional data that was generated, and training without additional data. Followings are the studies under each category.

2.2.2.1 Training with additional data

The first group of methods includes those that use additional training data. This study [159], Open Set Back Propagation (OSBP), suggests a technique for training a feature generator and a classifier that involves labeling unlabeled target samples as unknown and combining them with labeled source samples. This approach trains a feature generator to extract features that distinguish known target samples from unknown. Training this generator moves target samples away from the boundary and leads the probability of an unknown target sample to deviate from the pre-defined threshold. These features are then taken by a classifier to place a separator between source and target data and output the probability of target samples to reject unknowns.

Another research regarding document classification was reported by Shu et al. [171]. This classifier is the combining of a joint open classification with a sub-model. Seen and unseen classes are distinguished and rejected respectively by an open classification model. The sub-model finds the relation between two given samples to be identified if they belong to the same or different classes. Additionally, the number of invisible classes of the rejected samples can be obtained by considering this sub-model as a distance function for clustering.

Recently, Dhamija et al. [39] combined SoftMax with the Entropic Open-Set and Objectosphere losses considering the background and unknown training samples. These losses increase SoftMax entropy for unknown inputs while minimizing the Euclidean length of deep representations of unknown samples. This modification increases separation in deep feature space and improves the handling of background and unknown classes.

The Open Deep Network (ODN) that Shu et al. [172] suggested uses the work labelled "unknown data". It requires numerous manually added annotations. In particular, it added a new column to the weight matrix that corresponds to the unidentified category and initialized it as:

$$W_{N+1} = \alpha \frac{1}{N} \sum_{n=1}^{N} W_n + \beta \frac{1}{M} \sum_{m=1}^{M} W_m \tag{2.4}$$

Where W_n is the weight column for the n^{th} category that is known. Additionally, ODN included another phrase W_m which is the weight columns of M greatest activation values to emphasize the related recognized categories because they should be important in the initialization. To support the new category, the transfer weight W and the W_{N+1} are concatenated. ODN additionally introduces multi-class triplet thresholds (accept threshold, reject threshold, and distance threshold) to detect new categories. The index of a sample's top confidence value must be greater than the acceptable threshold in order for it to be accepted as a labeled class, and only then. If all of the confidence values fall below the rejected threshold, a sample would be regarded as unknown. If the gap between the top and second maximal confidence values is more than the distance-threshold, samples between the accept threshold and reject threshold would also be allowed as a labeled class.

In order to discern between anomalous and in-distribution occurrences, [73] introduced the Outlier Exposition (OE). OE used data that was ”out-of-distribution” (OOD), represented as $\mathcal{D}_{out}$ from other datasets. Target samples are being designated as $\mathcal{D}_{in}$ and are currently ”in-distribution.” The model is then taught signals to look for and heuristics to learn to identify which dataset a query samples. The objective function of OE has the following representation given a model f and the initial learning objective $\mathcal{L}$:

$$\mathrm{E}_{(x,y)\sim\mathcal{D}_{in}}[\mathcal{L}(f(x), y) + \lambda \mathrm{E}_{x'\sim D_{in}^{out}}[\mathcal{L}_{OE}(f(\acute{x}), f(x), y)]] \tag{2.5}$$

$\mathcal{D}_{out}^{OE}$ is a dataset of outlier exposure. The equation shows that for both ”in-distribution” and ”out-of-distribution” data, the model strives to minimize the objective $\mathcal{L}$. The paper also utilized the cross-entropy maximum softmax probability baseline detector for $\mathcal{L}_{OE}$. Additionally, $\mathcal{L}_{OE}$ was set to a margin ranking loss on the log probabilities $f(\acute{x})$ and f when labels are not provided. However, the OOD dataset that is selected affects how well this strategy works.

2.2.2.2 Training with additional data that was generated

This sub-category of open-set recognition approaches includes the research projects that produce additional training data. Recently, there exist abundant research on OSR based on the scheme of Generative Adversarial Networks (GANs) [62]. A GAN which recently stands out among various deep neural networks consists of a generator and a discriminator. Generally, the generator produces synthetic samples and the discriminator learns to decide if a sample is obtained from the generator or the real dataset. G-OpenMax [58] extends OpenMax in adversarial settings and applies GANs to generate unknown instances. These synthetic instances are utilized as an extra training label apart from known labels to adjust the classifier and estimate the probability of unknown classes. The proposed data augmentation technique which is applied to two datasets of hand-written digits and characters has shown itself to be an enhancement of the unknown class identification. However, using it over natural images does not show any performance improvement due to the difficulty of generating plausible images with respect to the training classes as candidates to represent unknown classes.

Along with a similar motivation, another GAN-based approach which is more effective than G-OpenMax for OSR was proposed by Neal et al. [121]. This strategy, which is called counterfactual image generation, searches for synthetic images by adopting an encoder-decoder GAN technique. These images, referred to as counterfactual-images, are a member of unknown classes but they look like known classes. Using the GAN framework, another work [85] aims at generating synthesis data which were served as fake unknown classes for the classifier to make it robust against real unknown classes. Yu et al. [203] proposed the adversarial sample generation (ASG) framework that produces unseen class data. Besides neural networks, ASG can be applied to several learning modes. Inspired by GANs, Yang et al. [200] proposed a novel model called Open-GAN. In this model, fake target samples are constructed from the generator automatically. Afterwards, the discriminator is modified to adapt multiple classes together with an unknown class.

This work [98] added two additional terms to the original cross-entropy loss. The first one (confident loss) forces out-of-distribution samples to achieve less confident predictions by the classifier. And the second one (adversarial generator) is for generating the most effective out-of-distribution samples for the first one. Unlike the original generative adversarial network(GAN), which generates samples similar to in-distribution samples, the proposed generator generates "boundary" samples in the low-density area of in-distribution acting as out-of-distribution samples. Finally, they jointly train the confident classifier and adversarial generator to make both models improve each other.

There is also recent interest in exploiting deep neural networks for applying OSR to text classification ([140, 141, 66, 170, 171, 36, 189]). Inspired by OpenMax, Prakhya et al. [140] developed an incremental convolutional neural network (CNN)-based text classifier. In contrast to OpenMax, which applies a single mean activation vector, this approach finds the k medoids of every trained class. Compared to image classification, they believe this method represents a class more accurately due to a much smaller number of classes. Then, the distances between the class activation vectors and the corresponding k class medoids are calculated. Applying the average of the k distances, a Weibull model is made for every training class that returns a probability of inclusion of the respective class. Open-set probability is then defined by subtracting the sum of all inclusion probabilities (total closed-set probability) from 1. A sample is either labeled as unknown, if the total open-set probability exceeds the maximum closed-set value, or assigned the class with the highest closed-set probability. Another work [189] combined CNN classification and three outlier detection methods to analyze the output vector of CNN and identify an unknown class.

A DOC (deep open classifier) proposed by Shu et al. [170] is a variant of the CNN

[92] architecture for text classification. This method was compared with OpenMax and represents better performance. One issue related to OpenMax is classifying samples which are difficult to handle as these samples are mostly classified as members of unknown classes. The proposed classifier addresses this issue where the SoftMax layer is replaced by a one-vs-rest final layer of sigmoid activations. The network is trained using a novel loss function to perform joint classification and unknown detection and reduce the open space risk. They show that the risk of open space is reduced further for rejection and the algorithm is improved further by tightening the sigmoid functions' decision boundaries with Gaussian fitting. One possible drawback of this approach would be the lack of compact abating property of the sigmoids which may cause the problem of unbounded open space risk when they are activated by an infinitely distant input from all of the training data.

2.2.2.3 Training without additional data

The OpenMax [12] proposed by Bendale et al. in 2016 was the first deep open-set classifier without using background samples. Since then, few deep open-set classifiers have been reported. OpenMax does not directly focus on the recognition of adversarial inputs, although it supports the rejection of fooling and unknown images. Rozsa et al. [155] compared DNNs using the traditional Softmax layer with Openmax on their robustness to adversarial examples. Although Openmax is more robust than Softmax to adversarial examples and outperforms networks with thresholding SoftMax, it does not provide robustness to sophisticated adversarial construction techniques. This work adapts the concept of Meta-Recognition [163] on activation vectors to formally solve OSR for image classification. Initially, a Neural Network undergoes training with a conventional Softmax layer to minimize cross-entropy loss. Subsequently, the activation vector for each training instance is determined, from which the per-class mean activation vector (MAV) is derived. Following this, the distance of each training instance from its respective MAV is computed, and individual Weibull distributions are fitted to a certain number of the largest distances for each class. Finally, the values of the activation vector are adjusted based on the probabilities derived from the Weibull distribution, and these adjusted values are aggregated to denote the activation value for the unknown class. The class probabilities, now encompassing the unknown class, are then computed using Softmax on the newly adjusted activation vector.

Another methodology for OSR is using a weightless neural network, denominated WiSARD [4]. Compared to various classifiers, WiSARD does not rely on prior knowledge regarding data distribution, which is usually unavailable in OSR tasks. The proposed model assigns fitness scores to each class and evaluates how well a given observation matches the previously stored knowledge. This classifier applies such a fitting level for rejection according to the similarity rating and proximity between corresponding features. Computing score thresholds, this paper [19] developed a rejection-capable WiSARD to identify whether observations pertaining to the class with the highest score or the best score is below the defined threshold, and then it is considered as an outlier. Following that, after proposing some exploratory results, a fully developed methodology is detailed in [20]. This paper investigates how to adapt the WiSARD classifier for OSR by carrying out detailed distance-like calculations and defining the rejection thresholds at the training.

Until recently, almost all existing deep open-set techniques included standard neural networks which are trained in a closed set environment and different activations which are analyzed to infer unknowns. However, relying on discriminative features of known classes in such systems causes specialization of learned representations to known classes and is not useful to represent unknowns. In contrast, some approaches enhance the learned representation to keep useful information to jointly perform known classification and unknown detection. Classification-Reconstruction learning for Open-Set Recognition (CROSR) is the novel framework proposed by Yoshihashi et al. [201] most recently. This is the first neural network architecture which involved hierarchical reconstruction blocks and trained networks for joint classification and reconstruction of input samples. The proposed system consists of a closed-set classifier which exploits learned prediction y for known class classification, and an unknown detector which uses a reconstructive latent representation z together with y for unknown detection, where y and z are provided by training a deep net. In this technique, utilizing reconstruction of input samples from low-dimensional latent representations [76], allows unknown detectors to exploit a wider pool of features that may not be discriminative for known classes. This study which considers deep representation learning is similar to [207] in terms of sharing the idea of reconstruction-based representation learning; however, [207] uses a single layer linear representation.

Recently, Oza et al. [126] combined a shared feature extractor that provides a latent space representation of an input image, along with a decoder and a classifier. While the decoder and classifier both take the latent representation as the input, the output of the decoder is the reconstructed image and that of classifier is the label of image. After training all networks and accomplishing both classification and reconstruction tasks, the reconstruction error tail from the known classes is modeled utilizing EVT to enhance the performance. Reconstruction errors from the decoder network are utilized to reject samples from the unknown classes. Another work [125] proposed an algorithm using class conditional auto-encoders. In this method, the training procedure is divided into two parts to improve the learning of open-set identification scores. The first part, closed-set classification, is learned by an encoder using the traditional classification loss and the closed-set training setting, while a decoder reconstructs conditioned on class identity to train an open-set identification model and accomplish the second part of the training. Furthermore, EVT is used to model reconstruction errors and obtain the operating threshold.

In a recent paper [101], the known and unknown classes are first distinguished based on entropy measurement and training the model on a modified cross entropy loss by dedicating a low and high cross-entropy for known and unknown classes, respectively. Then it uses the weighted square difference loss to assign unlabeled target samples to known classes based on the likeliness. Another work [46] used CNN to extract effective features, along with a rejection approach depending on the uncertainty metric Breaking Ties [109] to build a recognition method. During the rejection scenario, for a given test sample, the class confidence scores are computed. After that, the difference between the first and second best scores are used as an indicator to recognize an unknown sample. If this value goes over a pre-determined threshold, the observation is recognized as a known sample.

Chapter 3

Proposed Methodology

In this chapter, we introduce the proposed methodology, which is based on the concept of replacing the original feature space representation with one that is more useful for open set recognition. This transformed space can be learned by the proposed formulation of a novel loss function. This represents a significant contribution of our research, presented herein. Before we delve into the details of the methodology, we will first introduce the concept of activation vectors and illustrate their analysis and visualization with examples. Activation vectors provide a space in which we define our target criteria and effectively demonstrate their capability in distinguishing between instances of known and unknown classes. Additionally, we will explore the OpenMax algorithm, a pioneering approach in Open Set Recognition, which has inspired our proposed methodology.

3.1 Interpretation of Activation Vectors

An artificial neural network is composed of the neurons or processing-computing units which are interconnected to each other and organized in three types of layers called input, hidden, and output layers. The input layer receives data and communicates to the hidden layer(s) where the actual processing is done using the weighted paths. Then, the hidden layer(s) connect to the last layer in the network to give the output. A three-layer neural network including an input layer L_1, a hidden layer L_2, and an output layer L_3 is shown in Figure 3.1. Neurons represented by the circles communicate with each other by sending signals over several weighted connections. Every neuron represents a specific output function called the activation function f. The circles labeled X_1, X_2 and X_3 represent the inputs fed in the forward direction through the network and the bias unit is represented by label b. Weights W are basically the effect of previous layers' neurons on the ones of the current layer.

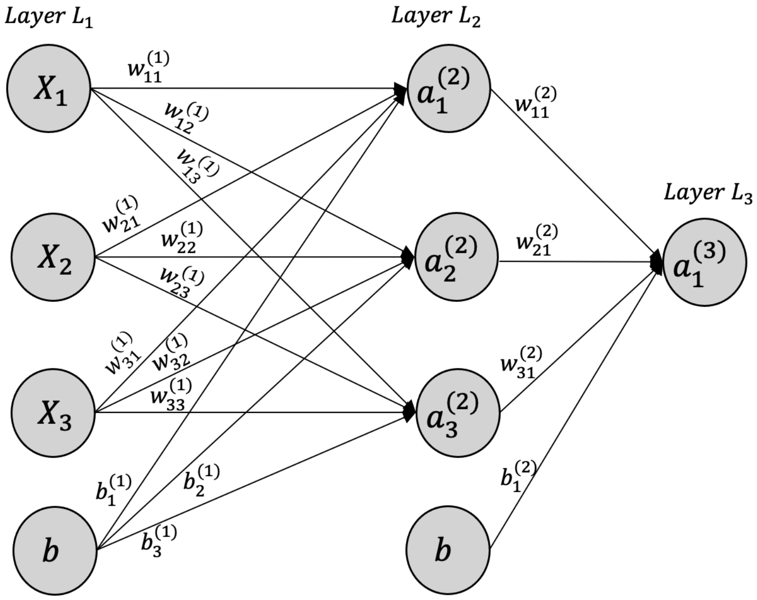


Figure 3.1: Three-layer neural network.

In forward propagation process which is the transformation of data from the input layer to the output layer, first a weighted sum of inputs (the linear transformation of weights w.r.t to inputs) are calculated and passed to the activation function. Then, the state of activation that is the output of the neuron goes to the next layer. The activation values of neurons in one layer act as the input for the activation function of the next layer, where the weight connections define the amount of this contribution. If the activation value of the ith unit in lth layer is defined as a_i^l, for the given inputs and weights shown in Figure 3.1, the function output can be calculated as follows:

$$\begin{aligned}
a_1^{(2)} &= f(W_{11}^{(1)}X_1 + W_{21}^{(1)}X_2 + W_{31}^{(1)}X_3 + b_1^{(1)}) \\
a_2^{(2)} &= f(W_{12}^{(1)}X_1 + W_{22}^{(1)}X_2 + W_{32}^{(1)}X_3 + b_2^{(1)}) \\
a_3^{(2)} &= f(W_{13}^{(1)}X_1 + W_{23}^{(1)}X_2 + W_{33}^{(1)}X_3 + b_3^{(1)}) \\
a_1^{(3)} &= f(W_{11}^{(2)}a_1^{(2)} + W_{21}^{(2)}a_2^{(2)} + W_{31}^{(2)}a_3^{(2)} + b_1^{(2)})
\end{aligned}$$

This is how the activation values of different layers are updated in the forward propagation process. The activation function f is a mathematical function which is used to nonlinearize the neural network. Sigmoid, hyperbolic tangent(tanh), ReLU and Softmax are four commonly used activation functions.

After forward propagation process, it's time to calculate the loss (prediction error) which is the difference between the actual output and predicted one. The method to calculate the loss is called loss function. Backward propagation is a mechanism to calculate the gradient of the loss function with respect to the neural network's weights and update the value of weights to minimize prediction error. Generally, the network is trained in the backward propagation and the objective of training is about finding weights that minimize prediction error for each of the training examples. This process leads to a set of properly adjusted weights that enables the neural network to be used effectively for the purpose it is initially designed for.

In most deep networks, the final fully-connected layer's output is processed by the SoftMax function to generate a probability distribution across K predefined class labels. In a conventional multi-class classifier, all inputs are classified into one of the known classes observed during training (see Figure 3.2). The SoftMax layer plays a vital role in this closed set assumption. It acts as a gradient-log-normalizer for the categorical probability distribution, making it a common choice for the final fully connected layer in neural networks. Deep networks produce scores in the second-to-last layer, known as the activation vector, through convolutional operations. Denoting the activation levels of sample X for each class as $AV(X) = a_1(x), a_2(x), ..., a_k(x)$. After training, an input X yields an activation vector $AV(X)$, and the SoftMax layer calculates:

$$P(y = j|X) = \frac{e^{a_j(X)}}{\sum_{i=1}^{K} e^{a_i(X)}} \tag{3.1}$$

where the denominator sums over all classes to ensure the probabilities over all classes sum to 1. In the multi-class classification, the cross-entropy loss function quantifies the difference between the SoftMax output and the desired result, aiming to minimize classification errors during training. Finally, during the testing time, the SoftMax function provides the probability that input X belongs to class y, $y = 1, ..., K$. However, the closed set nature of deep networks can lead to misclassification when assigning unknown samples to the class with the highest SoftMax score.

In this study, our focus is on the network values obtained from the penultimate layer, which corresponds to the fully connected layer preceding the SoftMax function. These values are responsible for extracting higher-level representations from the input data. We refer to these values as the activation vector ($\overrightarrow{AV}$). For open set images, the activation vector typically exhibits a small magnitude. This small magnitude can be attributed to the absence of the unknown class during the training phase, which hinders the network's ability to learn its distinctive features. By leveraging the characteristics of this layer, we incorporate the information derived from the $\overrightarrow{AV}$ into our approach.

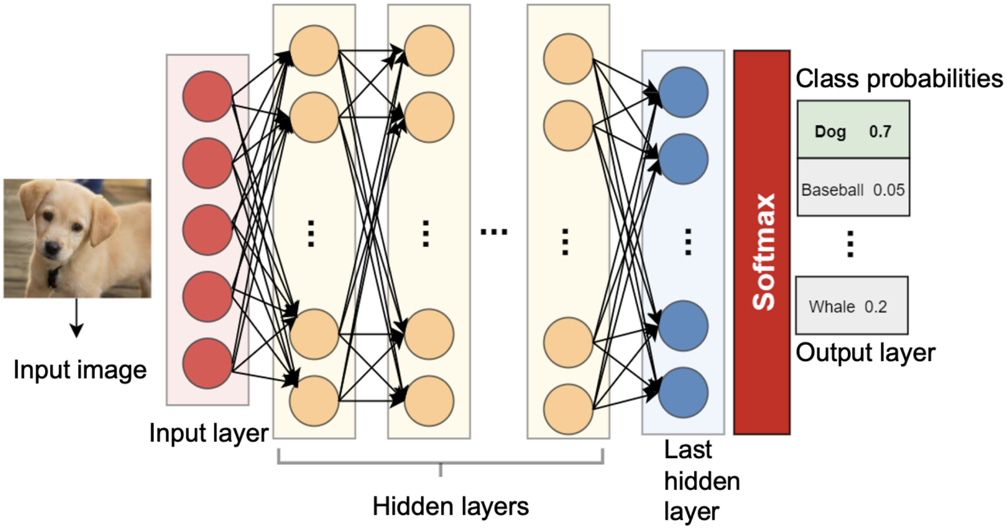


Figure 3.2: Schematic structure of DNNs. Example of an image classification task taking an image and outputting the confidence scores for a predefined set of classes.

Figure 3.3 presents activation vectors representing the response patterns of a trained deep neural network for 9 known classes for the CIFAR-10 dataset, alongside an open set class Truck. These AVs are extracted from the penultimate fully connected layer of the trained model and are depicted as color pixels, each offering a unique insight into the model's recognition capabilities. Figure 3.4 represents the visualization of the 9 hidden nodes in the last hidden layer, denoted as $h_1, h_2, \ldots, h_9$, for each of the 9 known classes in the CIFAR-10 dataset. Figure 3.3 showcases individual AVs for various images, with distinct AVs being delineated by black lines. Each input image is transformed into an AV, portrayed as color pixels. The horizontal axis represents the deep network's activation energy, while the vertical axis signifies the response to specific classes. Ranges for categories such as Airplane, Automobile, Bird, and others are marked on the left side of the image. A close examination of the AVs reveals patterns of activation where related classes often exhibit correlated responses. For instance, Deer and Dog share several visual features, resulting in correlated responses, while less correlation exists with classes like Automobile or Frog. These AVs accentuate the differences between the response patterns for open set images and the model's AVs. These distinctions demonstrate the potential for using them to enable deep networks to support open-set recognition.

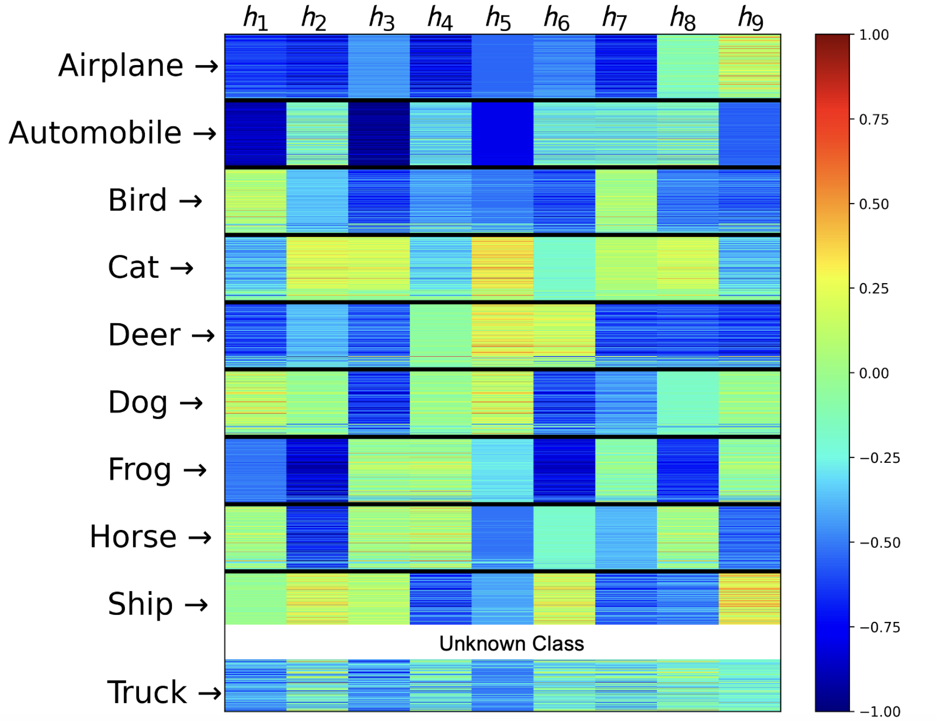


Figure 3.3: Visualization of activation vectors for CIFAR-10 classes and an Open Set class Truck.

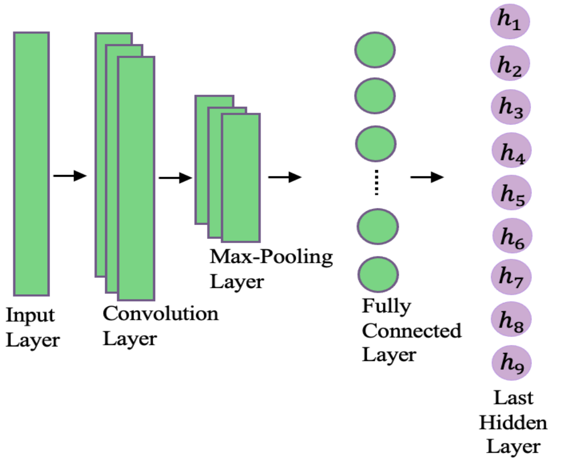


Figure 3.4: Visualization of the 9 nodes represented in Figure 3.3 as $(h_1 ... h_9)$ in the last hidden layer for 9 known classes of CIFAR-10 dataset.

3.2 Understanding OpenMax: Inspiration for Our Proposed Method

OpenMax is a state-of-the-art algorithm in the realm of open set recognition which introduces a novel approach for handling open set scenarios by recalibrating the confidence scores of predicted classes. In this section, we delve into its key components and underlying mechanisms, and highlight how these insights have influenced the development of our novel approach.

The OpenMax algorithm introduces a new layer, called the OpenMax layer, which recalibrates the softmax scores produced by the softmax layer of a neural network. This recalibration involves adjusting the softmax scores not only for known classes but also for an additional unknown class. In traditional classification tasks, the output of the penultimate layer is often treated as independent per-class score estimates. Each value in this layer corresponds to the likelihood or confidence of the input belonging to a specific class. However, OpenMax takes a different perspective. Instead of viewing these per-class scores independently, OpenMax considers them as providing insights into the distribution of related classes. This means that the activation values in the penultimate layer are not just individual confidence scores but rather representations of how the input data is distributed across different classes or categories. Initially, a neural network undergoes training with the conventional Softmax layer to minimize cross-entropy loss. After training, the activation vector of each training instance is computed and, using these activation vectors the per-class mean of the activation vector is calculated for each class separately over only correctly classified training examples. Then, a Weibull distribution is fitted to each class to model the distribution of distances based on the η largest distances between all correct training instances and their associated per-class mean (see Figure 3.5). By fitting a Weibull distribution to these distances, OpenMax captures the variability and distribution characteristics of the correct instances. This information is utilized to compute the OpenMax score, which represents the likelihood that an input belongs to a known class or an unknown class. The fitting of the Weibull distribution involves estimating its parameters, such as shape and scale, from the selected distances. These parameters characterize the distribution of distances for each class. Let $\rho_j = (\tau_j, \lambda_j, \kappa_j)$ be an estimation of parameters for class j. Where τ, λ and κ are the location, shape and scale parameters of the Weibull distribution.

During testing, each test sample goes through the OpenMax score calibration process as the following. Having ρ as a vector of parameters for each class, the Weibull CDF probability is used on the distance between a test sample x and per-class mean of activation vector for the core of the rejection estimation. They expect the EVT function of distance to provide a meaningful probability only for few top ranks. Thus, weights are computed for the α largest activation classes and is used to scale the Weibull CDF probability as:

$$\omega_i(x) = 1 - \frac{\alpha - i}{\alpha} e^{-\left(\frac{\left\| x - \tau_{(i)} \right\|}{\lambda_{(i)}}\right)^{k_{(i)}}} \quad i = 1, ..., \alpha \tag{3.2}$$

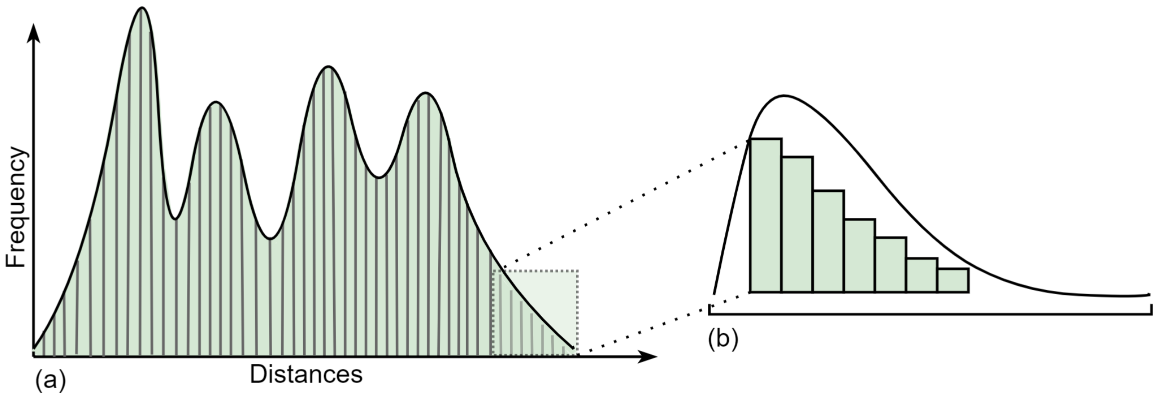


Figure 3.5: (a) Class particular distance distribution based on measured distance between each correctly classified training example and the associated per-class mean of activation vector. (b) A Weibull distribution related to a certain number of the largest such distances which is fitted separately for each class.

The revised activation vector is then computed with the top scores changed as:

$$\hat{V}\left(x\right) = V\left(X\right)\omega(X) \tag{3.3}$$

A pseudo-activation is computed for the unknown class, keeping the total activation level constant as:

$$\hat{V_0}\left(x\right) = \sum_i V_i\left(x\right)\left(1 - \omega_i(x)\right) \tag{3.4}$$

The class probabilities (now including the unknown class) are then calculated using Softmax on the new redistributed activation vector as:

$$\hat{P}\left(y = j \mid X\right) = \frac{e^{\hat{V_j}(x)}}{\sum_{i=0}^{N} e^{\hat{V_i}(x)}} \tag{3.5}$$

Finally, the class with the maximum over all probabilities is the predicted class:

$$y^* = argmax_j P(y = j|x) \tag{3.6}$$

For convenience they define the unknown class to be at index 0. So, OpenMax provides probabilities that support explicit rejection when the unknown class (y = 0) has the largest probability. This maximum probability is then subject to the uncertainty threshold to support the rejection of uncertain inputs as well:

$$\textit{Reject input if} \quad y^* == 0 \quad \textit{or} \quad P(y = y^*|x) < \epsilon \tag{3.7}$$

The limitation of OpenMax is that this technique does not enhance the feature representation to facilitate better detection of unknown samples. OpenMax employs the standard cross-entropy loss function during neural network training, which may not lead to class instances being consistently projected around their respective means. In an ideal scenario, class instances would be tightly clustered around their respective class means in the feature space. This would make it straightforward to use the distance from the mean as a measure for classifying samples. However, in practice, neural networks trained with cross-entropy optimize model parameters to minimize overall error across the dataset, rather than explicitly encouraging tight clustering around class means. Consequently, class instances may be dispersed or spread out in the feature space. Therefore, using the distance from the per-class mean as a criterion for detecting unknown samples may not always be accurate. Moreover, because the testing distance function is not used during training (Euclidean distance), it might not necessarily be the right distance function for that space.

3.3 Superlative Loss Function

The training phase of the proposed algorithm is depicted in Figure 3.6. In this figure, the activation vector ($\overrightarrow{AV}$) of each training sample belonging to the K known classes are visualized, with distinct colors assigned to each of the K known classes. The algorithm begins by computing the means of the known classes as an initial step. To accomplish this, the mean activation vectors ($\overrightarrow{MAV}$) for each class are calculated by averaging the $\overrightarrow{AV}$ values of the training instances associated with that class:

$$\overrightarrow{MAV}_i = \frac{1}{N_i}\sum_{n=1}^{N_i}\overrightarrow{AV}_{i,n} \qquad (3.8)$$
$$1 \le i \le K$$

In the above equation, for the K known classes, N_i is the number of training examples in each class. $\overrightarrow{AV}_{i,n}$ represents the activation vector of each training sample n in the i-th known class, where $1 \le n \le Ni$ and $1 \le i \le K$.

We utilize principal component analysis (PCA) to extract the principal components from the mean activation vectors of each class. This enables us to efficiently reduce the dimensionality of the data while preserving essential information encapsulated in these components. The three highest-ranking principal components, denoted as PC_1, PC_2, and PC_3, are selected. Employing a restricted number of principal components not only reduces the number of variables in the optimization process of declaring superlative space, leading to less time consumption but also ensures that all features have the desired impact based on their significant portion of the total variance. Additionally, it enhances data exploration and visualization efficiency through dimensionality reduction, while also tackling the curse of dimensionality issue in high-dimensional spaces [2]. This leads to more robust distance calculations that remain resilient against noise distortion. It is worth mentioning that, we compute principal components once before training for all examples. We save the coefficients corresponding to the top three principal components as shown in yellow box in Figure 3.6. This precomputation reduces the need for repeated PCA calculations during training. During the training phase, for each iteration of a batch comprising 256 samples, we avoid the recalculations of principal components. Instead, we leverage the precomputed coefficients of the top three principal components. This involves a simple matrix-vector multiplication, eliminating the need for repetitive principal component computations. Consequently, the features are represented in the feature space by three coordinates. In our approach, each class is represented by a single point, denoted as $\overrightarrow{PM}$, which captures its projection onto the three selected principal components.

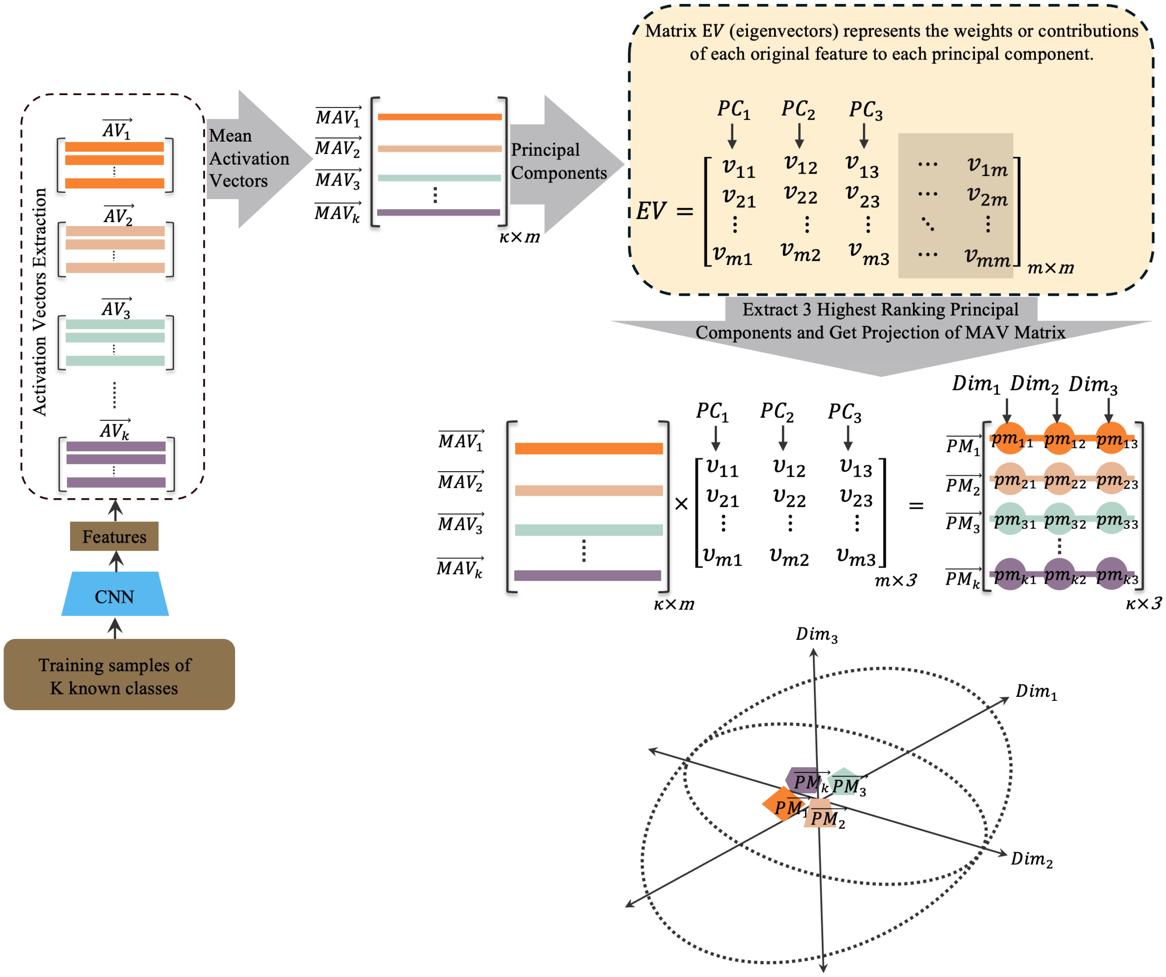

Activation Vectors Extraction
$\overrightarrow{AV_1}$
$\overrightarrow{AV_2}$
$\overrightarrow{AV_3}$
$\overrightarrow{AV_k}$
Features
CNN
Training samples of
K known classes
Mean
Activation
Vectors
$\overrightarrow{MAV_1}$
$\overrightarrow{MAV_2}$
$\overrightarrow{MAV_3}$
$\overrightarrow{MAV_k}$
$\kappa \times m$
Principal
Components
Matrix EV (eigenvectors) represents the weights or contributions of each original feature to each principal component.
PC_1 PC_2 PC_3
$EV = \begin{bmatrix} v_{11} & v_{12} & v_{13} & \cdots & v_{1m} \\ v_{21} & v_{22} & v_{23} & \cdots & v_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & v_{m3} & \cdots & v_{mm} \end{bmatrix}_{m \times m}$
Extract 3 Highest Ranking Principal Components and Get Projection of MAV Matrix
$\times \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ \vdots & \vdots & \vdots \\ v_{m1} & v_{m2} & v_{m3} \end{bmatrix}_{m \times 3} =$
Dim_1 Dim_2 Dim_3
$\overrightarrow{PM_1}$ pm_{11} pm_{12} pm_{13}
$\overrightarrow{PM_2}$ pm_{21} pm_{22} pm_{23}
$\overrightarrow{PM_3}$ pm_{31} pm_{32} pm_{33}
$\overrightarrow{PM_k}$ pm_{k1} pm_{k2} pm_{k3}
$\kappa \times 3$
Dim_3
Dim_1
Dim_2
$\overrightarrow{PM_k}$
$\overrightarrow{PM_3}$
$\overrightarrow{PM_1}$
$\overrightarrow{PM_2}$

Figure 3.6: An overview of acquiring the three highest-ranking principal components and the projection of the Mean Activation Vector for each class.

After completing all the defined steps and variables, our primary objective is to maximize the distance between the represented points ($\overrightarrow{PM}s$) within the feature representation. To achieve this, we establish a boundary that encompasses these points.

In determining the radius of this boundary, we prioritize the first principal component, as it contributes the most to the overall variation. We subtract the maximum value of PC_1 from the minimum value of PC_1 across all known classes, which provides us with the maximum spread length. The hyperparameter Γ (where $\Gamma \in \mathbb{R}, \Gamma > 1$) in equation 3.11 effectively enhances the inter-class distance when combined with this spread length. In our experiments, after conducting a spatial study and visualizing the locations of the mean classes, we found that setting $\Gamma = 2$ accommodates both the absence of overlap between classes and sufficient spacing for placing unknown samples. The radius of the boundary denoted as R_b defined as the following.

$$PC_{1,max} = \max(\overrightarrow{Dim_1}) \tag{3.9}$$

$$PC_{1,min} = \min(\overrightarrow{Dim_1}) \tag{3.10}$$

$$R_b = \Gamma \times (PC_{1,max} - PC_{1,min}) \tag{3.11}$$

Subsequently, our objective is to maximize the distance between points in the space. For this purpose, we have defined the following characteristics, which are applied to each batch of training samples during the network training process to achieve the ideal space we are aiming for. The first characteristic involves minimizing the distance between each point's current position $\overrightarrow{PM}$ and the boundary, compelling them to move closer to it. This process ensures maximum separation between points.

During the training of the network, this process is repeated for each batch of 256 samples. For each batch, we undertake the same process to obtain the $\overrightarrow{MAV}$. Then, we use the saved coefficients (matrix EV) to project each class into the three selected principal components and push each point towards the boundary. The overview of this process is shown at the top of the Figure 3.7. We refer to this characteristic as the "boundary distance,"denoted by BD:

$$BD = \sum_{i=1}^{K} (R_b - \|\overrightarrow{PM}_i\|_2)^2 \tag{3.12}$$

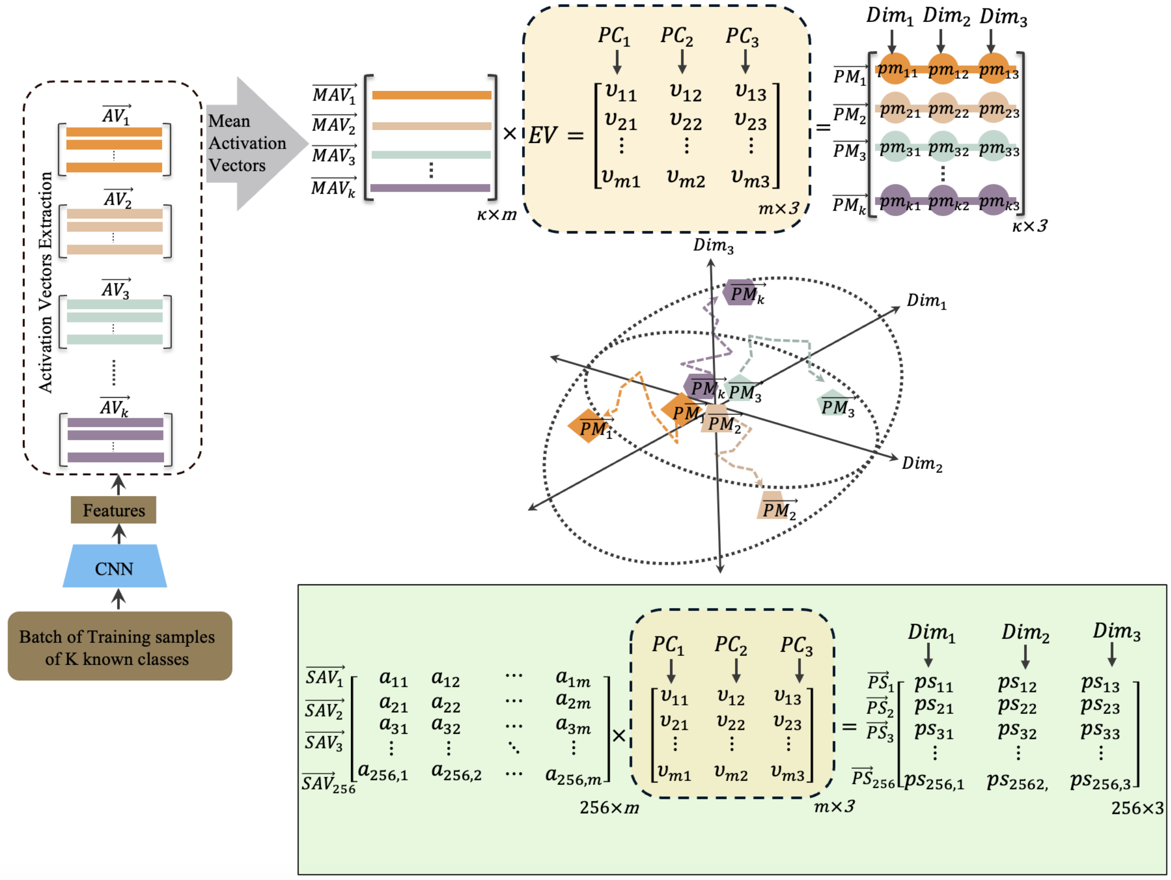

Activation Vectors Extraction
$\overrightarrow{AV_1}$
$\overrightarrow{AV_2}$
$\overrightarrow{AV_3}$
$\overrightarrow{AV_k}$
Features
CNN
Batch of Training samples of K known classes
Mean Activation Vectors
$\overrightarrow{MAV_1}$
$\overrightarrow{MAV_2}$
$\overrightarrow{MAV_3}$
$\overrightarrow{MAV_k}$
$\kappa\times m$
$\times$ $EV =$
PC_1 PC_2 PC_3
v_{11} v_{12} v_{13}
v_{21} v_{22} v_{23}
v_{m1} v_{m2} v_{m3}
$m\times 3$
$=$
Dim_1 Dim_2 Dim_3
$\overrightarrow{PM_1}$ pm_{11} pm_{12} pm_{13}
$\overrightarrow{PM_2}$ pm_{21} pm_{22} pm_{23}
$\overrightarrow{PM_3}$ pm_{31} pm_{32} pm_{33}
$\overrightarrow{PM_k}$ pm_{k1} pm_{k2} pm_{k3}
$\kappa\times 3$
Dim_3
$\overrightarrow{PM_k}$
Dim_1
$\overrightarrow{PM_k}$ $\overrightarrow{PM_3}$
$\overrightarrow{PM_3}$
$\overrightarrow{PM_1}$
$\overrightarrow{PM_1}$ $\overrightarrow{PM_2}$
Dim_2
$\overrightarrow{PM_2}$
$\overrightarrow{SAV_1}$ a_{11} a_{12} $\cdots$ a_{1m}
$\overrightarrow{SAV_2}$ a_{21} a_{22} $\cdots$ a_{2m}
$\overrightarrow{SAV_3}$ a_{31} a_{32} $\cdots$ a_{3m}
$\overrightarrow{SAV_{256}}$ $a_{256,1}$ $a_{256,2}$ $\cdots$ $a_{256,m}$
$256\times m$
$\times$
PC_1 PC_2 PC_3
v_{11} v_{12} v_{13}
v_{21} v_{22} v_{23}
v_{m1} v_{m2} v_{m3}
$m\times 3$
$=$
Dim_1 Dim_2 Dim_3
$\overrightarrow{PS_1}$ ps_{11} ps_{12} ps_{13}
$\overrightarrow{PS_2}$ ps_{21} ps_{22} ps_{23}
$\overrightarrow{PS_3}$ ps_{31} ps_{32} ps_{33}
$\overrightarrow{PS_{256}}$ $ps_{256,1}$ $ps_{2562,}$ $ps_{256,3}$
256×3

Figure 3.7: An overview of the proposed method and the necessary steps required to achieve the desired space.

During an experiment, we observed a situation where two points from different classes come into close proximity to each other as they approach the boundary. This undesired situation undermines the objective of maximizing the inter-class distances. To address this challenge, we introduce an additional constraint that ensures that the distance between each point and its nearest neighbor is maximized as they approach the boundary. This characteristic, known as "inter-separation", is denoted by IS and is defined as:

$$IS = \max_{1 \leq i \neq j \leq K} \|\overrightarrow{PM}_i - \overrightarrow{PM}_j\|_2^2 \tag{3.13}$$

By imposing this constraint, the algorithm ensures that points representing different classes maintain a sufficient separation. This helps to preserve the performance of the algorithm in terms of overall classification accuracy and maximization of inter-class distances. To further enhance the feature space and optimize performance, we aim to minimize the distance of each sample to its corresponding class mean. This is achieved by first obtaining the activation vector value (referred to as $\overrightarrow{SAV}$) for each sample in a batch of 256. Next, we multiply each $\overrightarrow{SAV}$ by the three highest-ranking predetermined coefficients (matrix EV) saved during the beginning of training. Subsequently, we project each sample (referred to as $\overrightarrow{PS}$) into a space defined by the three highest ranking components (see the green box in Figure 3.7). Finally, we calculate the distance of each sample to its corresponding class mean. The term "intra-compactness", denoted by IC, is used to describe this characteristic and defined as:

$$IC = \sum_{i=1}^{K} \sum_{n=1}^{N_i} \|\overrightarrow{PM}_i - \overrightarrow{PS}_n\|_2^2 \tag{3.14}$$

where $\overrightarrow{PS}_n$ represents the principal components of each of the N samples from the K known classes. By applying this equation, we compute the Euclidean distance between each sample and its corresponding class mean. The superlative loss, which is defined as a combination of the properties of boundary distance, inter-separation, and intra-compactness, is utilized to train the network. This loss function, denoted as $\mathcal{L}_s$, is minimized using mini-batch stochastic gradient descent with backpropagation, and described as:

$$\mathcal{L}_s = BD - IS + IC \tag{3.15}$$

Algorithm 1 presents a step-by-step explanation of the proposed method.

Algorithm 1: Training Phase

```
**Input:** $(X, Y)$: Training data and labels
**Require:** Activation levels in the penultimate layer $\overrightarrow{AV}_i = \overrightarrow{av_1} \ldots \overrightarrow{av_N}$ , Where $i = 1...K$ for $K$ known classes and $N$ is the number of samples in each class
**Require:** The mean activation vector $\overrightarrow{MAV}_i$
***coeff*** (internal $n \times m$ matrix with PAC coefficients, n is computed features and m is the number of principal components) $\leftarrow$ PCA($\overrightarrow{MAV}$)
$\overrightarrow{PM} \leftarrow$ Multiply $\overrightarrow{MAV}$ by the ***coeff*** PCA coefficient matrix
**Require:** $PC_{1,max} \leftarrow max(\overrightarrow{Dim_1})$, $PC_{1,min} \leftarrow min(\overrightarrow{Dim_1})$, ***coeff*** PCA coefficient matrix
**for** number of training iterations **do**
    Sample a mini-batch $(X_{batch}, Y_{batch})$ from $(X, Y)$
    **Compute:** $\overrightarrow{AV}_{batch,i} = \overrightarrow{av_1} \ldots \overrightarrow{av_N}$, Where $i = 1...K$
    **Compute:** The mean activation vector $\overrightarrow{MAV}_{batch,i}$
    $\overrightarrow{MP}_{batch,i} \leftarrow$ Multiply $\overrightarrow{MAV}_{batch,i}$ by the ***coeff*** PCA coefficient matrix.
    $BD \leftarrow (PC_{1,max},\ PC_{1,min},\ \overrightarrow{MP}_{batch,i})$
    $IS \leftarrow (\overrightarrow{MP}_{batch_i},\ \overrightarrow{MP}_{batch_j})$ Where $i \neq j$
    $\overrightarrow{PCA}_{batch} \leftarrow$ Multiply $X_{batch}$ by the ***coeff*** PCA coefficient matrix.
    $IC \leftarrow (\overrightarrow{MP}_{batch},\ \overrightarrow{PCA}_{batch})$
    $\mathcal{L}_s \leftarrow BD - IS + IC$
    Update parameters using stochastic gradient descent to minimize $\mathcal{L}_s$ loss
**end for**
```

3.4 Prediction for Known and Unknown Classes

During the inference phase of OSR, the main task is to classify a set of $K + 1$ labels into $K + 1$ distinct categories. Among these labels, the first K labels correspond to the known classes that the classifier has been trained on. The remaining label, $(K + 1)$st, is specifically assigned to represent the unknown class, indicating that an instance does not belong to any of the defined classes. In open set recognition, it is essential to set a threshold for uncertainty. This threshold serves as a criterion to distinguish between known classes and unknown ones. Without such thresholding, a deep neural network will always assign an open set image from an unknown category to the class that SoftMax identifies with the maximum response. While open set images may occasionally exhibit lower confidence, the maximum score will still lead to an assigned class. When comparing the activation vectors of the input image to the Mean Activation Vector (MAV) of the class to which it was assigned with the highest response, we often observe significant differences between the input activation vector and the MAV. In some cases, open set images may have activation responses closer to the AV of the assigned class, but the overall activation level remains low and may not be different enough to be rejected. This situation can arise when the input image belongs to a category closely related to a known class or when the object in the image is poorly defined, such as a small or indistinct object.

We adopted an approach where we estimate a separate threshold value for each class instead of applying a uniform threshold across all classes. Through our experiments, we observed that estimating a per-class threshold provides more accurate results. By tailoring the threshold to the specific characteristics of each class, we can effectively account for variations in the data distribution across different classes. To determine the class-specific threshold, we follow a specific procedure. This threshold value defines the minimum distance required between an instance and its nearest class mean for it to be classified as an unknown. After completing the training phase and updating the network's weights, we obtain learned superlative representation ($\overrightarrow{MAV}$) for each of the K known classes. Subsequently, we calculate the distances between $\overrightarrow{AV}$ of each training sample and its corresponding $\overrightarrow{MAV}$. For each individual class, these distances are sorted in ascending order to capture the largest of the distances. Then, we select the distance value at the 99th percentile as the class-specific threshold. During testing, we measure the distance between the $\overrightarrow{AV}$ of a test sample and the class means. This distance is then compared to the threshold value defined specifically for each class. If the distance exceeds the cutoff, it means that the sample is significantly further away from the class means. In this case, the sample is categorized as an unknown class and labeled as $K+1$. Conversely, if the distance falls below the threshold, it indicates that the test sample is relatively closer to one of the known classes. The final prediction is then determined by identifying the nearest class mean among the K known classes. The test sample is assigned the label corresponding to the class with the closest mean with the highest probability P:

$$y = \begin{cases} K+1, & \text{if distance} > \text{threshold} \\ \\ \underset{1 \leq i \leq K}{\operatorname{argmax}} \, P(y = i \mid \overrightarrow{x}), & \text{otherwise} \end{cases} \tag{3.16}$$

3.5 Summary

In this chapter, we propose a method for efficiently leveraging deep neural networks for open set recognition. The proposed methodology discussed in this chapter performs and improves the performance of DNN models in classification tasks. In Chapter 4, we present experimental results and comparisons between the proposed superlative loss and other methods. Additionally, in Chapter 5, we go beyond open set images and demonstrate the capability of the proposed model for handling adversarial examples.

Chapter 4

Experimental Analysis

First, this section will provide an introduction to the specifics of the dataset and the evaluation schemes. Next, we will outline the experimental setup and delve into the exploration analysis.

4.1 Implementation Details

Figure 4.1 represents the network architecture implemented for this experiment using TensorFlow, which consists of convolutional layers, max-pooling layers, and fully connected layers. To elaborate on how the input image progresses through the network: The initial input image, which is a 32x32x1 grayscale image, undergoes a series of operations to extract meaningful features. The first layer of convolution applies a set of 32 kernels, each with a size of 5x5 and a stride of 1, over the input image. These kernels are small-sized matrices that slide over the input image, performing element-wise multiplication at each position, and then summing up the results to produce a single value (scalar) in the output feature map. Each kernel captures different patterns or features from the input image like edges, textures, or shapes. We utilize the padding parameter set to 'SAME', which ensures that the input volume is zero-padded evenly at the borders, thereby enabling the output feature map to maintain identical spatial dimensions to the input. In contrast, employing 'VALID' padding results in no padding, leading to a reduction in the spatial dimensions of the output feature map compared to the input. By employing a kernel size of 5x5 in each convolution layer, the padding value can be calculated as $\left\lfloor \frac{filter_size-1}{2} \right\rfloor$. Thus, with a kernel size of 5x5, the padding value would be 2. Following this convolutional operation, a max-pooling layer with a kernel size of 2x2 and a stride of 2 reduces the spatial dimensions of the feature maps, effectively downsampling the extracted features while retaining important information. A typical max pooling operation involves dividing the input feature map into non-overlapping rectangular regions (usually 2x2 or 3x3), and taking the maximum value from each region. The second convolutional layer further refines the extracted features, employing 64 kernels with the same size (5x5) and stride (1) as the previous layer. Another max-pooling layer with identical parameters (2x2 kernel size, 2 stride) is then applied to downsample the feature maps generated by the second convolutional layer. The specific configuration of these layers, including kernel sizes, strides, and the number of kernels, can be observed in the provided diagram. After the convolutional layers, two fully connected layers are employed to further process the extracted features. The output of the last fully connected layer undergoes a Softmax function, resulting in the generation of a probability distribution across the known classes. In addition to the convolutional and fully connected layers, ReLU activation functions are applied to introduce non-linearity, and a Dropout mechanism with a keep probability of 0.2 is utilized to prevent overfitting in the fully connected layers. Furthermore, batch normalization is applied to all layers to improve network performance and stability. To train our networks effectively, we utilize the Adam optimizer with a learning rate of 0.001 for a total of 3000 iterations.

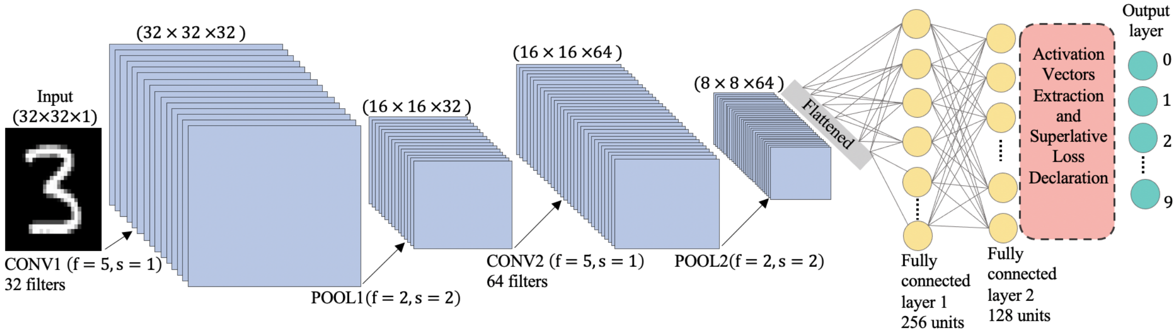


Figure 4.1: Visualization of the network architecture.

We assessed the performance of our approach using a set of four distinct datasets. **MNIST** consists of 70,000 gray scale images of handwritten numbers from 0 to 9. For each class, around 6,000 training samples and 1000 test samples are used.

Fashion-MNIST is a collection of gray scale images featuring 10 classes of clothing items. It comprises 60,000 training examples and 10,000 testing examples.

CIFAR-10 includes 60,000 32x32 color images distributed across 10 distinct classes, with each class containing 6,000 images. The dataset is divided into 50,000 training images and 10,000 test images. In our experimental setup, we transform the color images into gray scale.

In each dataset, we randomly select six classes as the known classes during the training phase, while the remaining classes are considered unknown during testing. As a result, we remove the instances belonging to unknown classes from the training set. However, the test set remains unchanged, containing both known and unknown class instances. This methodology enables us to simulate open-set recognition scenarios across all datasets. For each dataset, we create three distinct groups of open-set datasets referred to as $Set1$, $Set2$, and $Set3$. Each set consists of a random selection of six known classes, while the remaining four classes are designated as unknown classes. For example, the specific sets chosen for MNIST dataset for evaluation in this study are defined as $Set1 = [0, 2, 3, 4, 6, 9]$, $Set2 = [0, 1, 2, 5, 7, 8]$, and $Set3 = [0, 1, 3, 4, 7, 8]$. All other digits are considered as the unknown class in each set, respectively. We conducted an evaluation of five different approaches in our implementation framework. The first approach involves training a network, as shown in Figure 4.1, using the superlative loss ($\mathcal{L}_s$). As a baseline, the second approach focused on training a network using only cross-entropy loss (CE). The third approach is based on OpenMax (OM) ([11]), which we re-implemented using the original paper and the authors' source code. Considering that the superlative loss aims to enhance feature representation, it can be effectively combined with various other loss functions. In our study, we employed a combination of the superlative loss with both cross-entropy loss ($\mathcal{L}s + CE$) and OpenMax ($\mathcal{L}s + OM$) as fourth and fifth approaches, respectively. This setup involved training the network using the $\mathcal{L}s$ in conjunction with CE or OM. In the training process, first, the network weights are updated to minimize $\mathcal{L}(s)$, and subsequently, the weights are updated to minimize the other respective losses. To assess the performance, we conduct a total of 36 runs, with 12 experiments conducted for each of $Set1$, $Set2$, and $Set3$ for each of the five models. The findings, depicted in Figures 4.2, 4.3, and 4.4, show the average precision and recall for 12 runs across $Set1$, $Set2$, and $Set3$. These figures reveal that the performance of the combined models $\mathcal{L}(s)+CE$ and $\mathcal{L}(s)+OM$ are superior to the standalone versions, and we can observe reduced variance in these combined models as well.

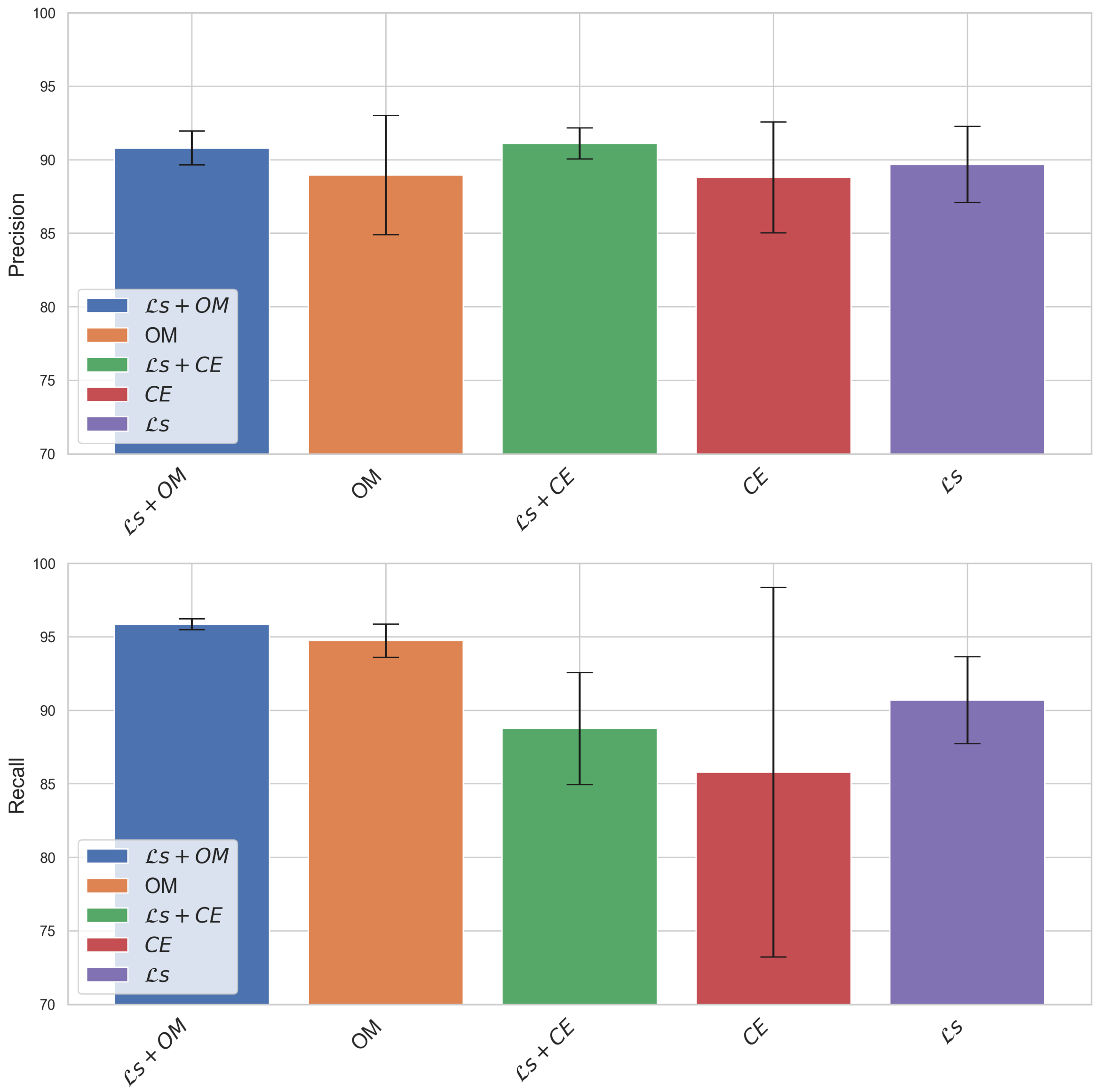


Figure 4.2: Average precision and recall with standard deviation error bars for $Set1$ over a total of 12 runs.

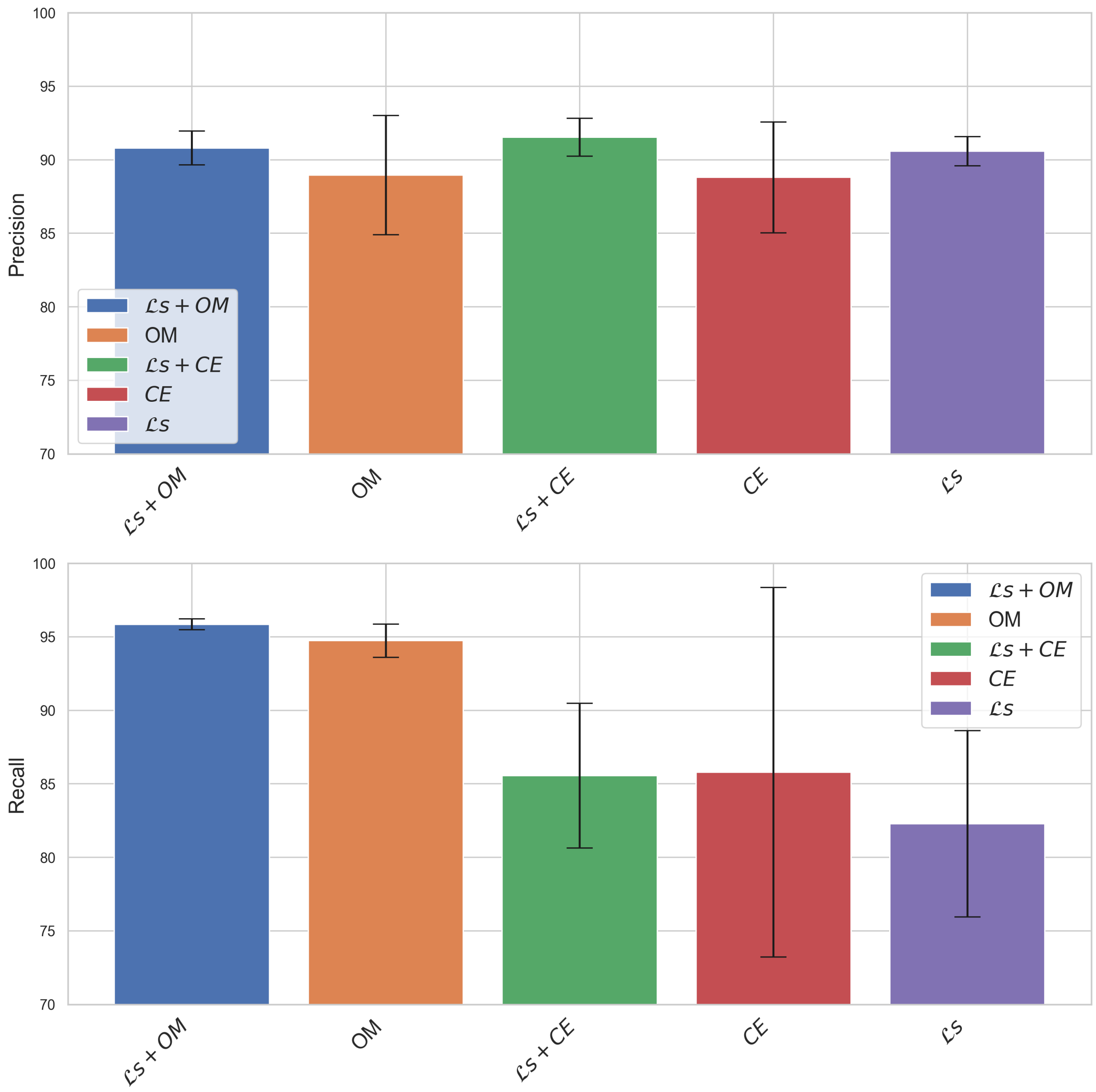


Figure 4.3: Average precision and recall with standard deviation error bars for $Set2$ over a total of 12 runs.

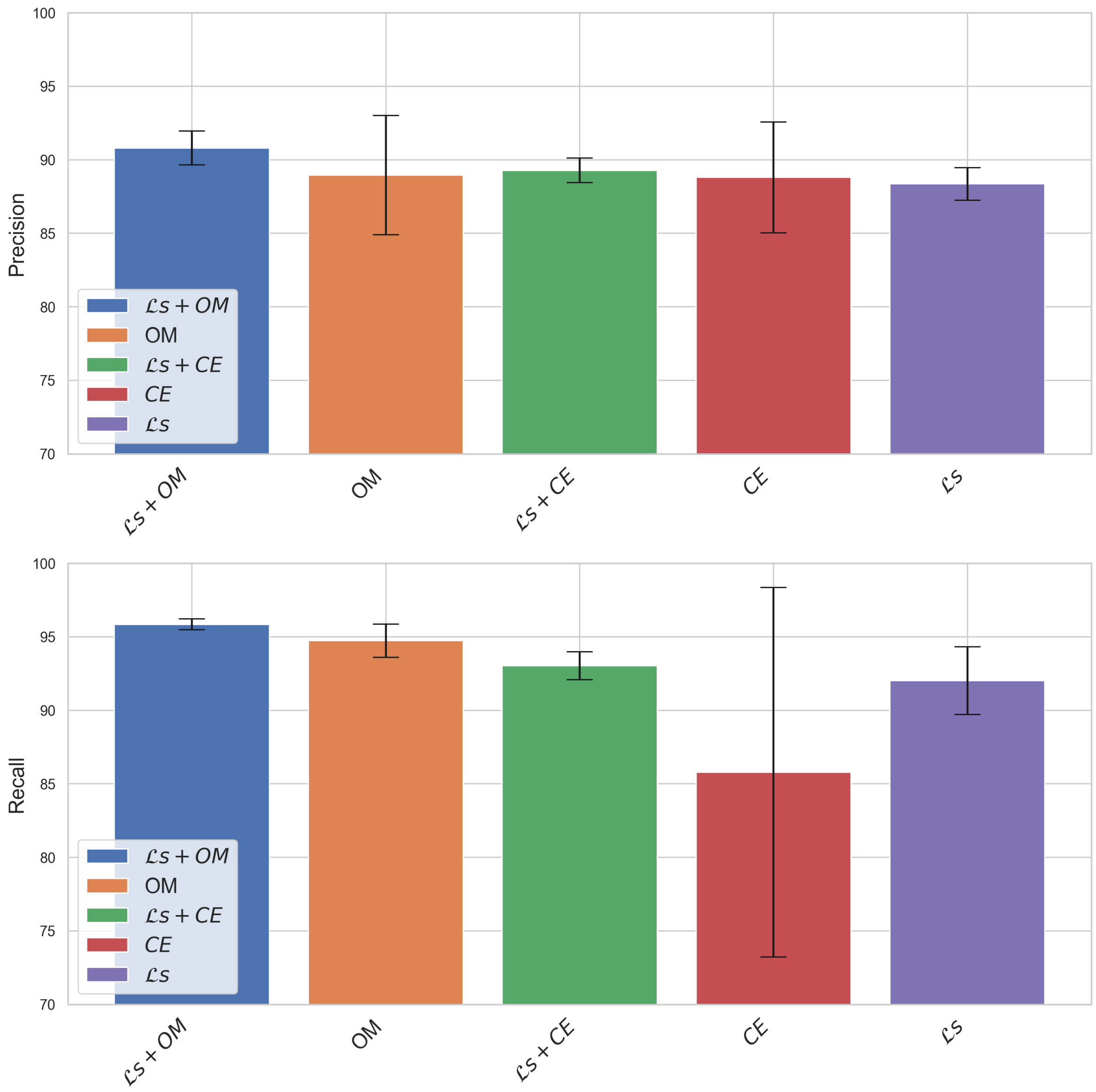


Figure 4.4: Average precision and recall with standard deviation error bars for $Set3$ over a total of 12 runs.

4.2 Performance Evaluation Metrics

To evaluate the performance of all the models across multiple classes, we employ precision, recall, F1 score, and accuracy. Precision measures the proportion of instances correctly classified as belonging to a specific class out of all instances predicted to belong to that class and it is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.1}$$

where TP (True Positives) represents the instances correctly classified as belonging to the class, and FP (False Positives) represents the instances incorrectly classified as belonging to the class. Recall assesses the classifier's ability to correctly identify all instances of a class out of all instances actually belonging to that class. It is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.2}$$

where FN (False Negatives) denotes instances of the class that were incorrectly classified as not belonging to the class. In multi-class classification, precision and recall are computed for each class individually, treating each class as positive in turn and the rest as negative. The mean precision and recall across all classes provide an aggregate measure of the classifier's performance. The F1 score is the harmonic mean of precision and recall, offering a balanced assessment of the classifier's performance. It is calculated as:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

The F1 score combines precision and recall into a single metric, providing a comprehensive evaluation of the classifier's effectiveness in multi-class classification tasks.

Tables 4.1, 4.2 and 4.3 present the evaluation results for the case of four unknown classes out of ten classes for MNIST, Cifar and Fashion-MNIST datasets. The average recalls, precisions, F1 scores, and accuracies are calculated for the K known classes and the unknown class and then averaged across the $K + 1$ classes to obtain the Overall values. As can be seen, $\mathcal{L}(s) + CE$ and $\mathcal{L}(s) + OM$ outperform their standalone versions. Figures 4.5, 4.6, and 4.7 provide a better visualization of these improvements. Specifically, combining CE with $\mathcal{L}s$ results in a significant improvement across all metrics such as F1 score and accuracy. Similarly, combining $\mathcal{L}s$ with OM leads to notable improvement in overall F1 score and accuracy. Importantly, the $\mathcal{L}s$ demonstrates enhanced robustness in detecting known and unknown classes compared to standalone CE, with a notable improvement of 10.33% in overall F1 score. This improvement underscores the effectiveness of the proposed method in detecting unknown classes while also reducing the classification error of the training data, thereby achieving competitive results compared to standalone CE in the classification task. This can be attributed to two key factors. Firstly, in the superlative representation, the learned activation vectors are more noticeable compared to conventional neural network features. Secondly, the superlative loss effectively guides the feature training process, enhancing the intra-class compactness and inter-class separation of the feature representation. Consequently, the combination of highly discriminative features and per-class thresholds contributes to a substantial enhancement in unknown detection performance.

Table 4.1: Comparison of average precisions, recalls, F1 scores, and accuracies across 36 runs for the MNIST dataset.

Methods	Precision			Recall			F1 Score			Accuracy
	Overall	Unknown	Known	Overall	Unknown	Known	Overall	Unknown	Known	Overall
CE	0.885	0.776	0.903	0.787	0.855	0.776	0.794	0.794	0.794	0.809
$\mathcal{L}s + CE$	0.906	0.858	0.914	0.891	0.861	0.896	0.89	0.857	0.895	0.883
OM	0.891	0.946	0.882	0.941	0.786	0.966	0.909	0.853	0.918	0.894
$\mathcal{L}s + OM$	0.898	0.97	0.886	0.956	0.797	0.982	0.921	0.873	0.929	0.908
$\mathcal{L}s$	0.895	0.852	0.903	0.883	0.838	0.891	0.876	0.84	0.882	0.87

Table 4.2: Comparison of average precisions, recalls, F1 scores, and accuracies across 36 runs for the Cifar dataset.

Methods	Precision			Recall			F1 Score			Accuracy
	Overall	Unknown	Known	Overall	Unknown	Known	Overall	Unknown	Known	Overall
CE	0.58	0.42	0.61	0.4	0.66	0.36	0.42	0.49	0.41	0.48
$\mathcal{L}s + CE$	0.61	0.49	0.62	0.58	0.6	0.6	0.59	0.51	0.6	0.56
OM	0.61	0.51	0.63	0.57	0.58	0.57	0.59	0.54	0.59	0.57
$\mathcal{L}s + OM$	0.63	0.57	0.64	0.67	0.53	0.7	0.65	0.53	0.67	0.61
$\mathcal{L}s$	0.56	0.64	0.57	0.62	0.41	0.68	0.57	0.45	0.6	0.54

Table 4.3: Comparison of average precisions, recalls, F1 scores, and accuracies across 36 runs for the Fashion-MNIST dataset.

Methods	Precision			Recall			F1 Score			Accuracy
	Overall	Unknown	Known	Overall	Unknown	Known	Overall	Unknown	Known	Overall
CE	0.711	0.757	0.703	0.798	0.435	0.859	0.732	0.542	0.764	0.689
$\mathcal{L}s + CE$	0.749	0.797	0.743	0.828	0.597	0.893	0.764	0.663	0.795	0.725
OM	0.715	0.797	0.702	0.816	0.423	0.882	0.74	0.539	0.773	0.698
$\mathcal{L}s + OM$	0.725	0.858	0.738	0.844	0.448	0.923	0.748	0.583	0.79	0.715
$\mathcal{L}s$	0.749	0.823	0.737	0.816	0.681	0.89	0.742	0.608	0.767	0.721

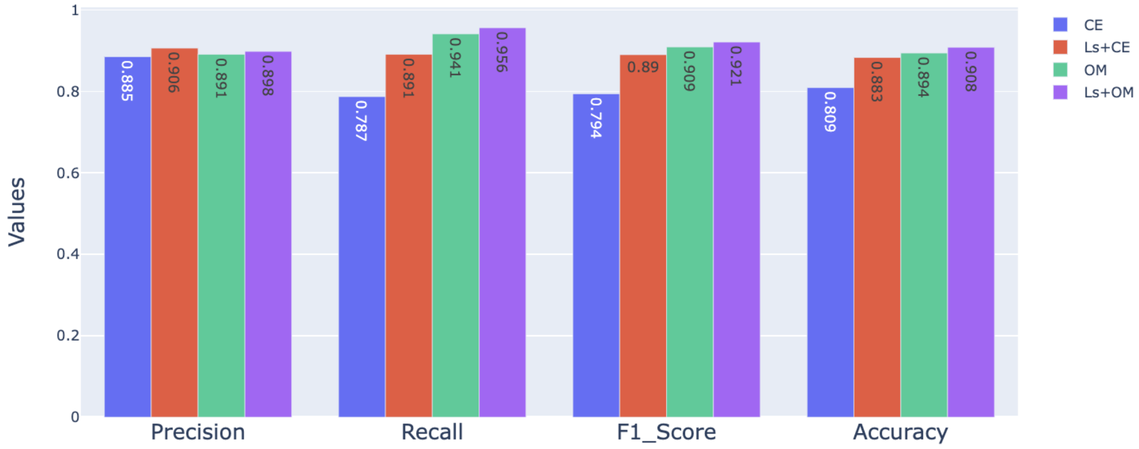


Figure 4.5: MNIST dataset overall metrics for standalone and combined methods for 36 runs.

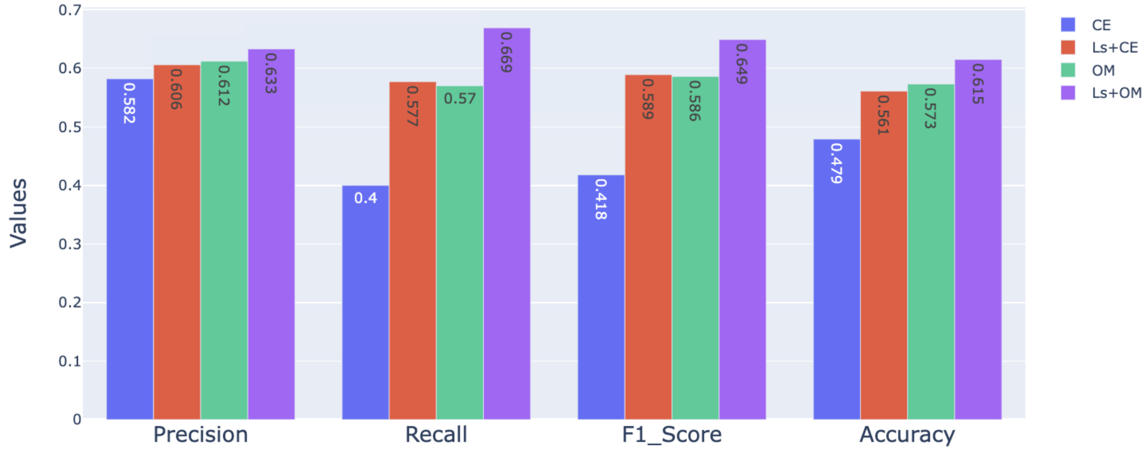


Figure 4.6: Cifar dataset overall metrics for standalone and combined methods for 36 runs.

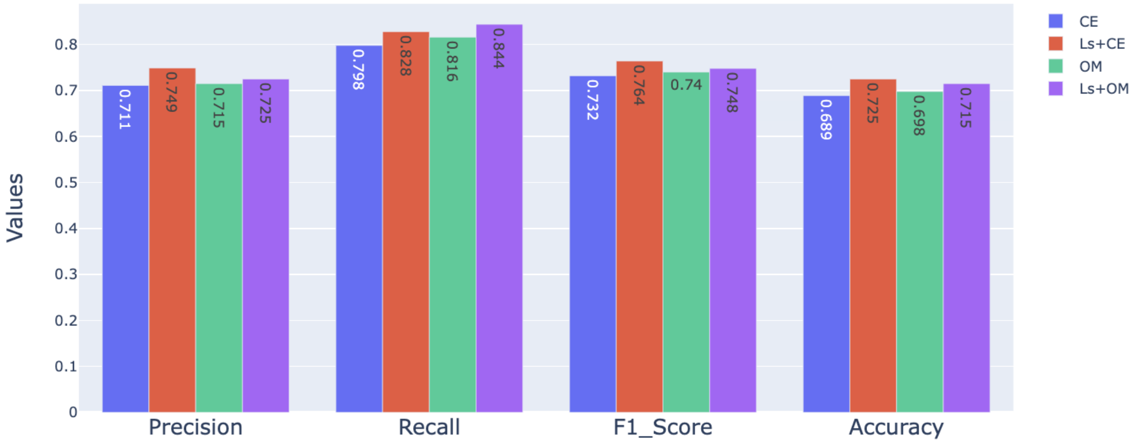


Figure 4.7: Fashion-MNIST dataset overall metrics for standalone and combined methods for 36 runs.

4.2.1 Statistical Analysis of Combined Models

In this subsection, we investigate the enhancement of performance metrics achieved by extending baseline models (CE and OM) with the superlative loss $\mathcal{L}s$. To quantify the improvements, we utilize the one-sample t-test on $\mathcal{L}s + CE$ and $\mathcal{L}s + OM$ models.

We compare the mean values of precision, recall, F1 score, and accuracy obtained from the baseline CE and OM models with sample data (36 runs values) from their respective extensions $\mathcal{L}s + CE$ and $\mathcal{L}s + OM$. The t-test measures how far the sample mean deviates from the population mean in terms of standard error units, providing insights into the statistical significance of the observed improvements. Tables 4.4, 4.5, and 4.6 present the results of the one-sample t-test for each performance metric, showcasing the calculated p-values. The blue highlighted spots indicate statistically significant improvements with a 95% confidence level. A smaller p-value indicates a higher level of significance in the observed improvement. If the p-value is smaller than a chosen significance level of 0.05, we reject the null hypothesis, providing strong evidence against it and suggesting a significant difference between the sample mean and the population mean. Therefore, a small p-value signifies that the observed difference between the sample mean and the population mean is unlikely to have arisen by random chance alone. Rejecting the null hypothesis in favor of the alternative hypothesis implies that there is a substantial difference between the two means, validating the efficacy of the model extensions.

Table 4.4: P-values for Statistical Significance Testing for the MNIST dataset

Methods	Baseline	Precision			Recall			F1 Score			Accuracy
		Overall	Unknown	Known	Overall	Unknown	Known	Overall	Unknown	Known	Overall
$\mathcal{L}s + CE$	CE	2.82e-10	3.38e-09	0.002	6.23e-15	0.430	9.92e-14	6.78e-19	1.84e-16	6.74e-19	3.08e-18
$\mathcal{L}s + OM$	OM	0.129	1.89e-20	0.461	4.58e-14	0.342	1.16e-18	0.0004	0.005	0.0001	0.0007
$\mathcal{L}s$	CE	0.003	1.10e-06	0.932	4.04e-11	0.130	2.35e-10	6.61e-13	2.73e-09	5.77e-13	1.94e-12

4.2.2 Efficiency Comparison of Models in Training Time

Table 4.7 presents a comparison of the average training times across 36 runs for the MNIST dataset with 3000 iterations. Despite the combined models of $\mathcal{L}s + CE$ and

Table 4.5: P-values for Statistical Significance Testing for the Cifar dataset

Methods	Baseline	Precision			Recall			F1 Score			Accuracy
		Overall	Unknown	Known	Overall	Unknown	Known	Overall	Unknown	Known	Overall
$\mathcal{L}s + CE$	CE	0.0002	1.37e-17	2.9e-06	9.19e-17	3.84e-14	5.54e-17	4.71e-19	0.9736	9.53e-19	2.09e-11
$\mathcal{L}s + OM$	OM	2.36e-05	5.13e-06	6.44e-07	2.37e-09	9.06e-16	2.38e-12	0.064	9.51e-10	0.0013	0.41101
$\mathcal{L}s$	CE	1.01e-09	3.5e-13	2.86e-10	0.0001	1.34e-24	3.98e-09	0.5261	7.14e-14	0.5519	0.0009

Table 4.6: P-values for Statistical Significance Testing for the Fashion-MNIST dataset

Methods	Baseline	Precision			Recall			F1 Score			Accuracy
		Overall	Unknown	Known	Overall	Unknown	Known	Overall	Unknown	Known	Overall
$\mathcal{L}s + CE$	CE	4.29e-14	9.77e-12	1.67e-13	2.1e-12	3.41e-17	2.9e-11	1.3e-13	1.1e-19	2.9e-12	1.6e-18
$\mathcal{L}s + OM$	OM	0.0115	1.62e-27	2.48e-12	1.14e-13	0.0275	1.57e-16	0.0194	2.83e-05	9.09e-06	2.08e-10
$\mathcal{L}s$	CE	1.88e-08	1.21e-17	3.95e-06	3.97e-05	2.99e-07	1.41e-07	0.0081	4.41e-08	0.55	1.31e-14

$\mathcal{L}s + OM$ showcasing superior accuracy and F1 score performance compared to standalone versions, the training times remain nearly identical, with only a marginal increase of approximately 1%, which is negligible. Moreover, comparisons between $\mathcal{L}s$ and CE models reveal that we can achieve even shorter training times without compromising performance. These findings underscore the efficiency of our proposed method in transforming samples to a more conducive space for open set recognition without incurring additional time overhead.

The results obtained from the training time, coupled with performance metrics, provide compelling evidence of the effectiveness of the proposed methodology. By conducting principal component analysis (PCA) once before training on all examples, we precompute and store the coefficients corresponding to the top three principal components. This precomputation significantly diminishes time consumption compared to algorithms that repeat PCA calculation during training.

Throughout the training phase, for each iteration of a batch comprising 256 samples, we circumvent the need for recalculating principal components. Instead, we leverage the precomputed coefficients of the top three principal components. This process entails a simple matrix-vector multiplication, effectively eliminating the requirement for repetitive principal component computations.

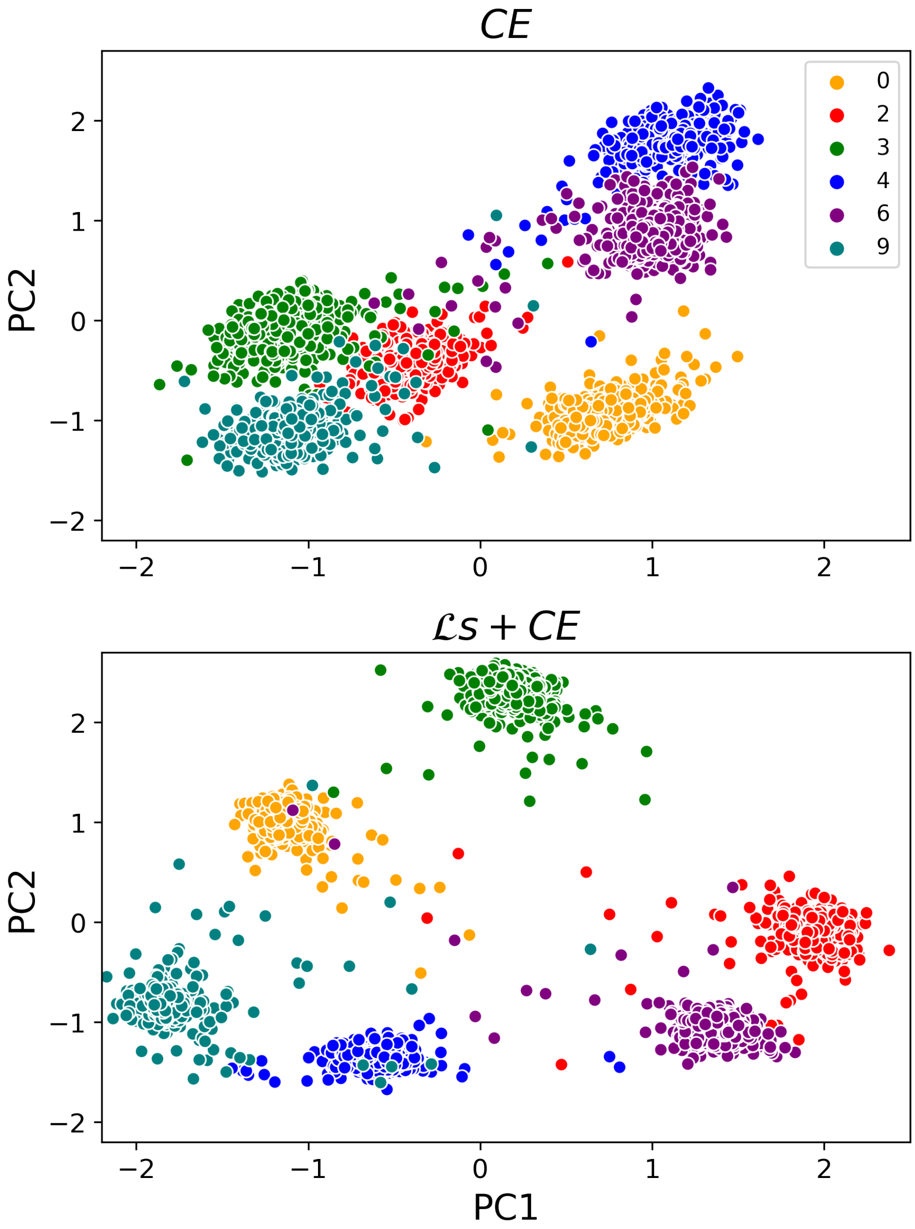
The utilization of PCA facilitates efficient data dimensionality reduction while retaining crucial information. Limiting the number of principal components not only simplifies the optimization process by reducing the number of variables but also ensures that all features contribute meaningfully based on their proportion of total variance. Consequently, this expedites data exploration and visualization while mitigating the challenges associated with the curse of dimensionality in high-dimensional spaces. As a result, more robust distance calculations are achieved, bolstering resilience against noise distortion.

Method	Training Time (seconds)
CE	1727
$\mathcal{L}s + CE$	3257
OM	1702
$\mathcal{L}s + OM$	1711
$\mathcal{L}s$	1725

Table 4.7: Training time for different methods

4.2.3 Visualization of Class Separation Enhancement

Figure 4.8 visually presents the superlative space of $Set1$ of MNIST dataset when subjected to the CE and $\mathcal{L}s + CE$ models in a 2D space. The x-axis corresponds to the first principal component (PC_1), while the y-axis represents the second principal component (PC_2). From the graph, it can be observed that the six classes associated with the CE model are roughly situated in the middle of the space. In contrast, the classes of the $\mathcal{L}s + CE$ model are positioned around them, exhibiting greater distances between the classes and greater compactness within each class individually. This transformation serves the purpose of the $\mathcal{L}s + CE$ approach, which aims to create more separation between the classes, benefiting closed-set classification and improving pen set recognition.


CE

$\mathcal{L}s + CE$

Figure 4.8: Feature space visualization of MNIST dataset in the experiments of CE vs $\mathcal{L}s + CE$. Labels 0,2,3,4,6,9 represent the known classes.

Moreover, Figure 4.9 illustrates the representation of unknown samples. It is notable that the unknown samples are positioned at the origin, where low probabilities are anticipated. In this depiction, we demonstrate that the combined model of $\mathcal{L}s+CE$ compactly clusters unknown samples at the origin even more effectively. Furthermore, it is important to mention that the proposed feature space transformation does not alter the placement of unknown samples; they remain stationary in the transformed space, unlike known samples which are relocated.

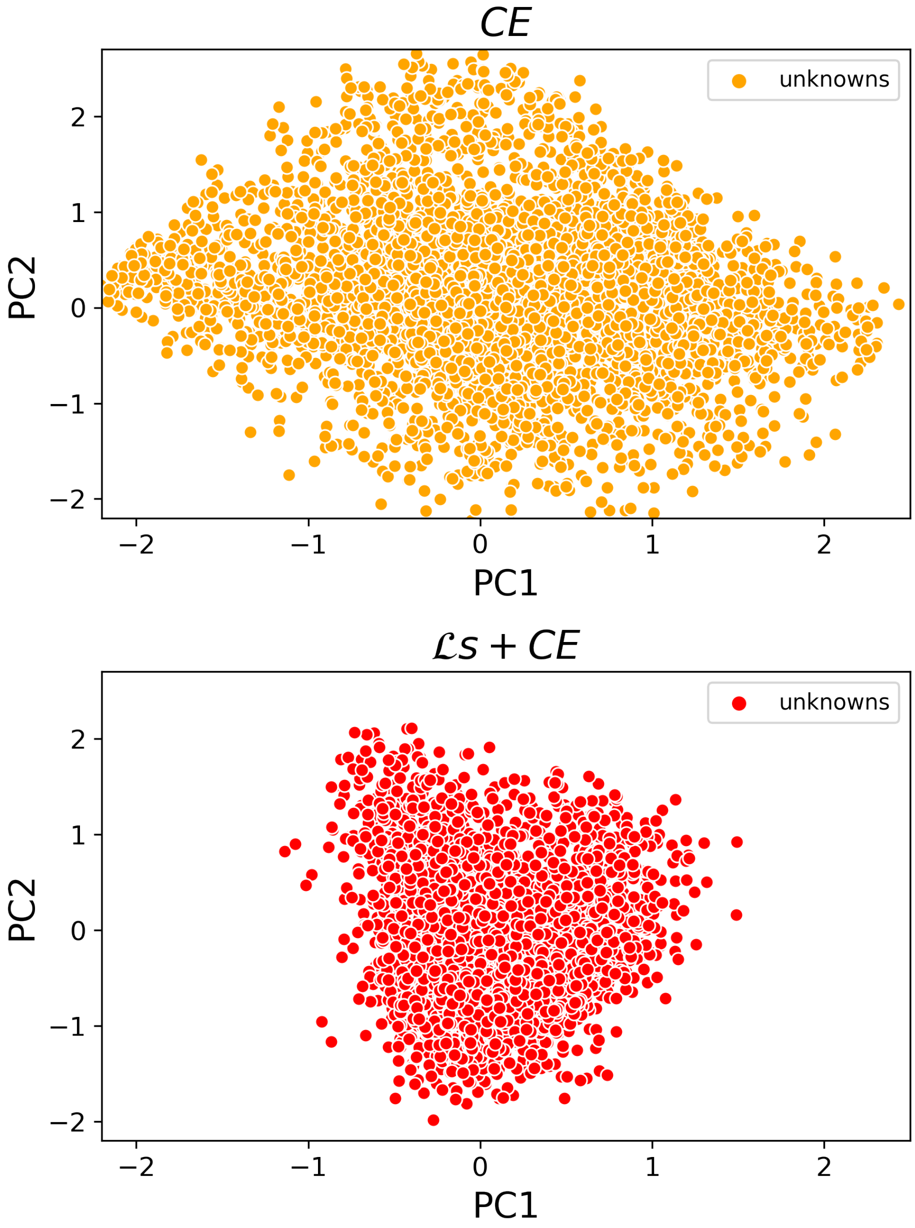


Figure 4.9: Representation of unknown samples of MNIST dataset and impact of feature space transformation.

As part of our investigation, we examined the relationship between the squared differences of the absolute feature values (feature magnitudes) of the six known classes and the unknown class. These feature values were obtained from the last fully connected layer before the softmax layer, demonstrating the spatial effectiveness of our proposed method in terms of Euclidean distance from features of unknown samples.

Figure 4.10 illustrates this relationship, with the x-axis representing the class index of the six known classes and the y-axis representing their corresponding squared differences from the unknown class. It is evident that both the blue line, representing the $\mathcal{L}s$ method, and the combined $\mathcal{L}s + CE$ method exhibit greater Euclidean distances to unknown samples across all six known classes, compared to the green line, which depicts the Euclidean distances of the six known classes to the unknown class in the CE model.

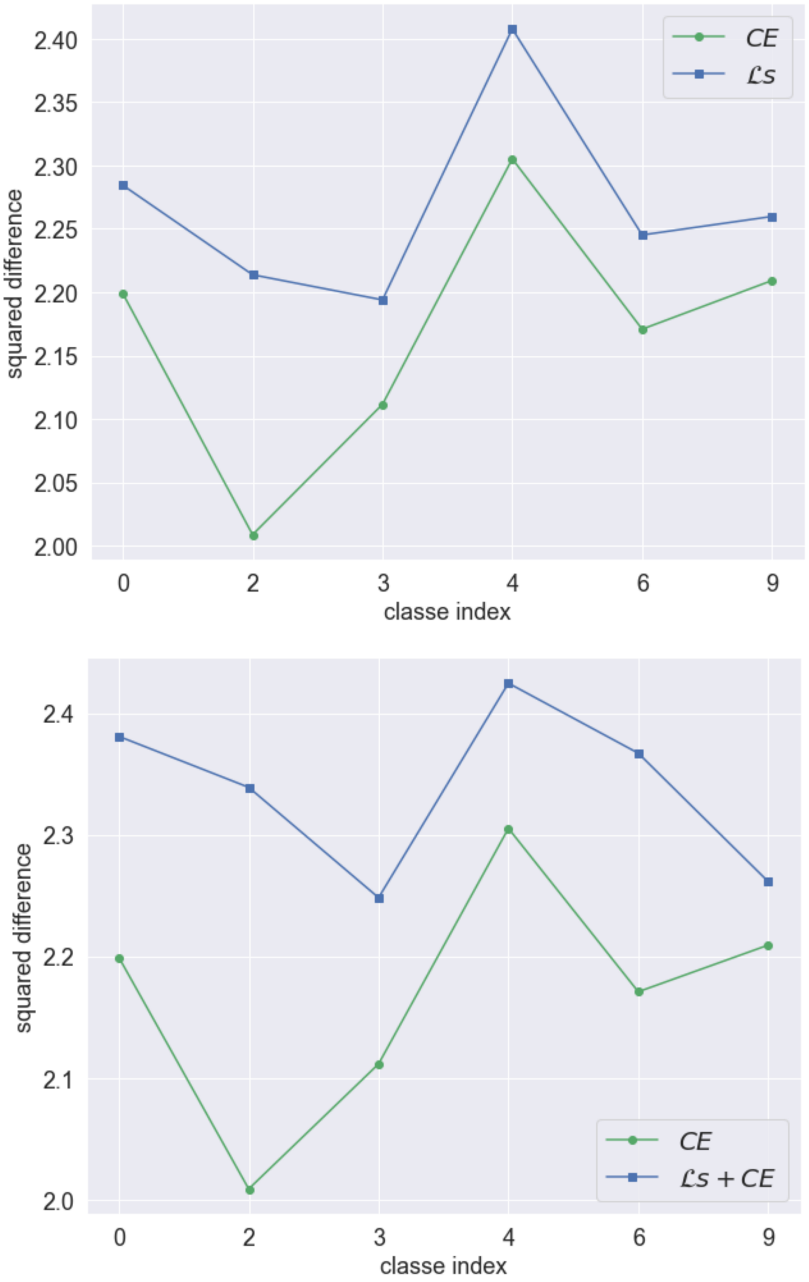


Figure 4.10: Squared differences of MAV values between the known and unknown classes.

We further investigate the value of the mean activation vector (MAV) of the unknown class in the last fully connected layer. The magnitude of these vectors is relatively small, as depicted in the last rows of Figure 4.11. This small magnitude can be attributed to the absence of the unknown class during the training phase, resulting in its features not being learned. By increasing the distance between the unknown and known classes, we noticed that the unknown classes were less affected, as evidenced by the heat maps of $\mathcal{L}s$ and $\mathcal{L}s + CE$ models. Despite the augmented separation, the MAV of the unknown class consistently remained close to the origin. This particular characteristic enables us to utilize the distance from the class mean as an outlier score, facilitating open set recognition.

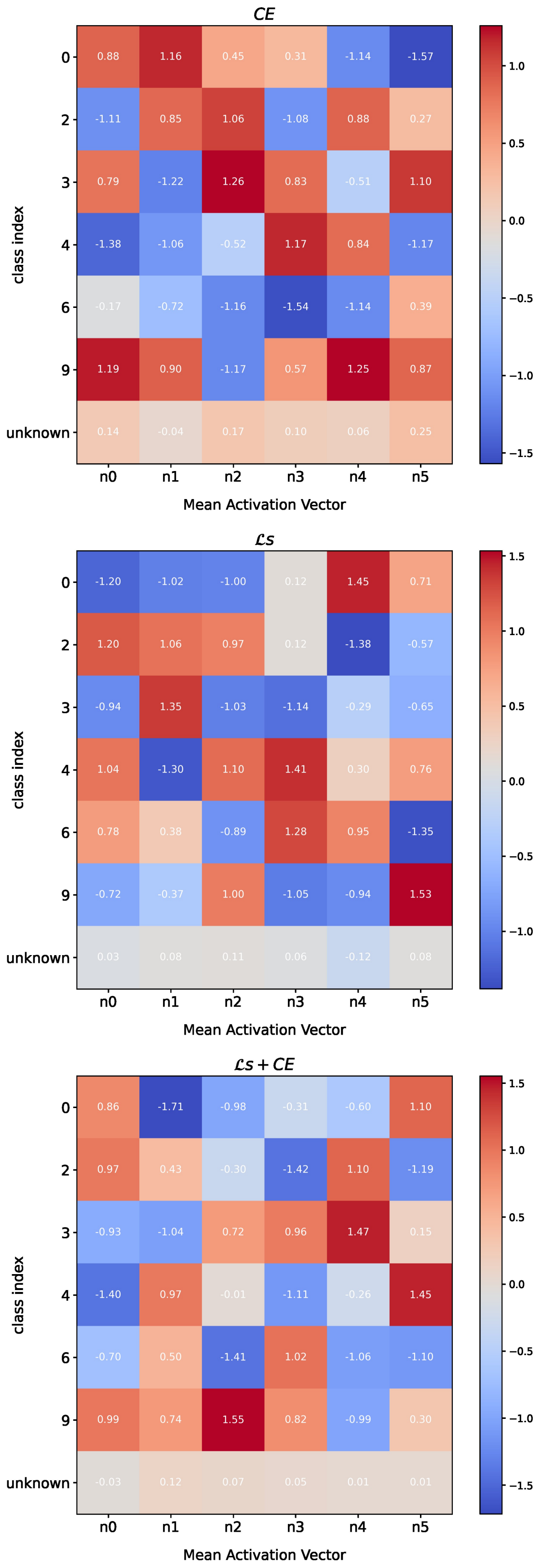


Figure 4.11: The heat map of mean activation vectors of the classes from the MNIST dataset.

4.3 Classification Accuracy Comparison

Figure 4.12 presents ROC curves and AUC scores for each of the five models to facilitate comparison. The evaluation was carried out by randomly splitting the initial dataset into 80% for training data and the remaining 20% for test data. This process was repeated 12 times, and the resulting average ROC curve is displayed. Upon comparing these curves, it becomes evident that both $\mathcal{L}s + CE$ and $\mathcal{L}s + OM$ outperform their standalone versions across all datasets.

For instance, when combining CE with $\mathcal{L}s$ on the Cifar dataset, the AUC score is 0.75, while the standalone CE achieves an AUC of 0.68. Similarly, combining $\mathcal{L}s$ with OM also leads to a notable improvement. Notably, $\mathcal{L}_s$ demonstrates a superior AUC score when compared to standalone CE.

4.4 Degree of Openness

We also investigate the model's performance under different degrees of openness ([161]), which is defined by considering the number of classes seen during training (C_{train}), the number of classes in the test set (C_{test}), and the number of classes to be identified during testing (C_{target}). The degree of openness is calculated using the following equation:

$$\text{openness} = 1 - \sqrt{\frac{2 \times C_{\text{train}}}{C_{\text{test}} + C_{\text{target}}}} \tag{4.4}$$

In this equation, C_{target} is defined as $C_{\text{train}} + 1$. Figure 4.13 displays the results obtained as the number of known classes varies. The x-axis represents the number of known classes (C_{train}), while the y-axis corresponds to the output of the above equation, indicating the degree of openness. For our experiments, we examine three degrees

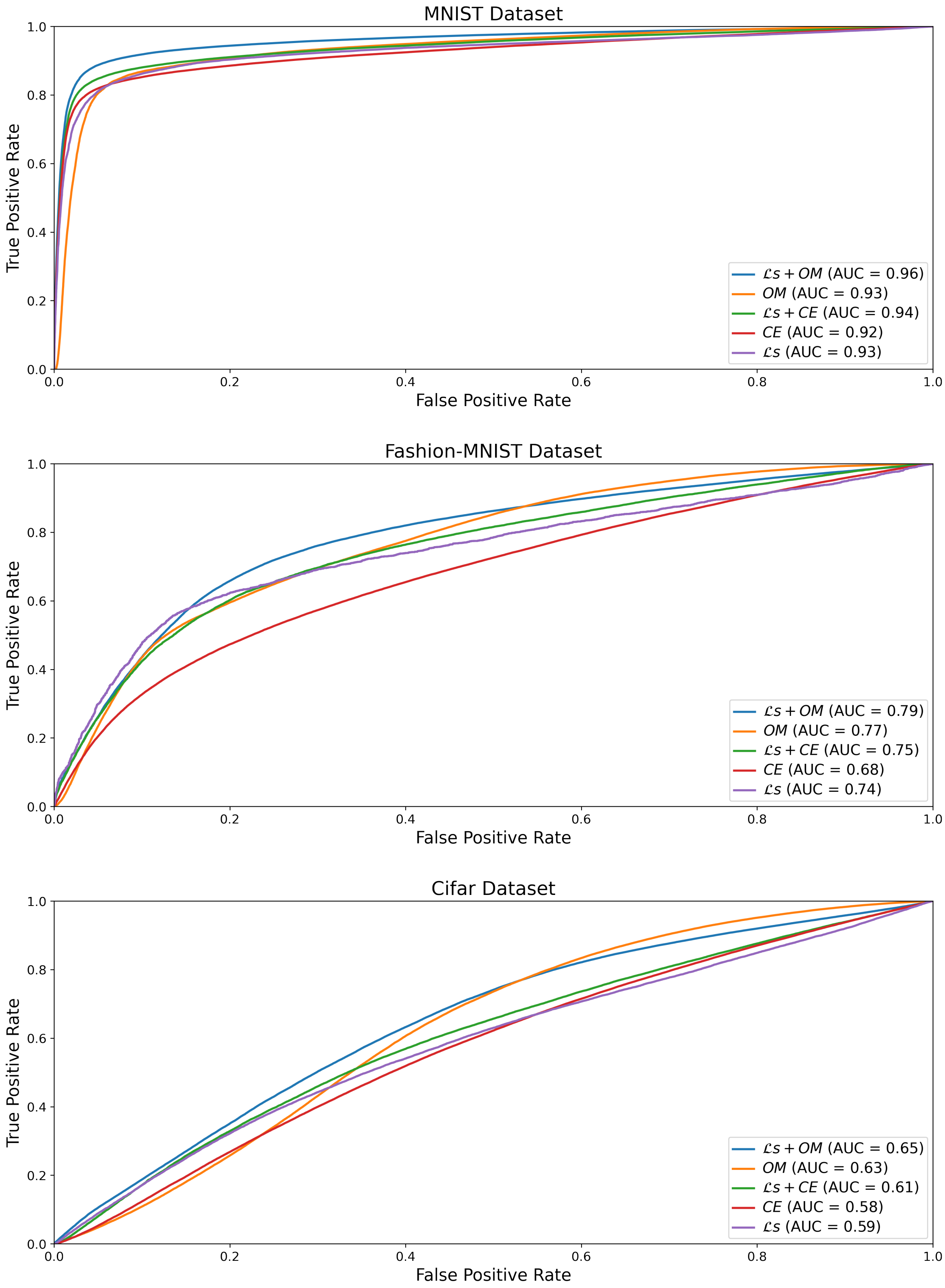


Figure 4.12: Recursive Operating Curve (ROC) comparison for three datasets.

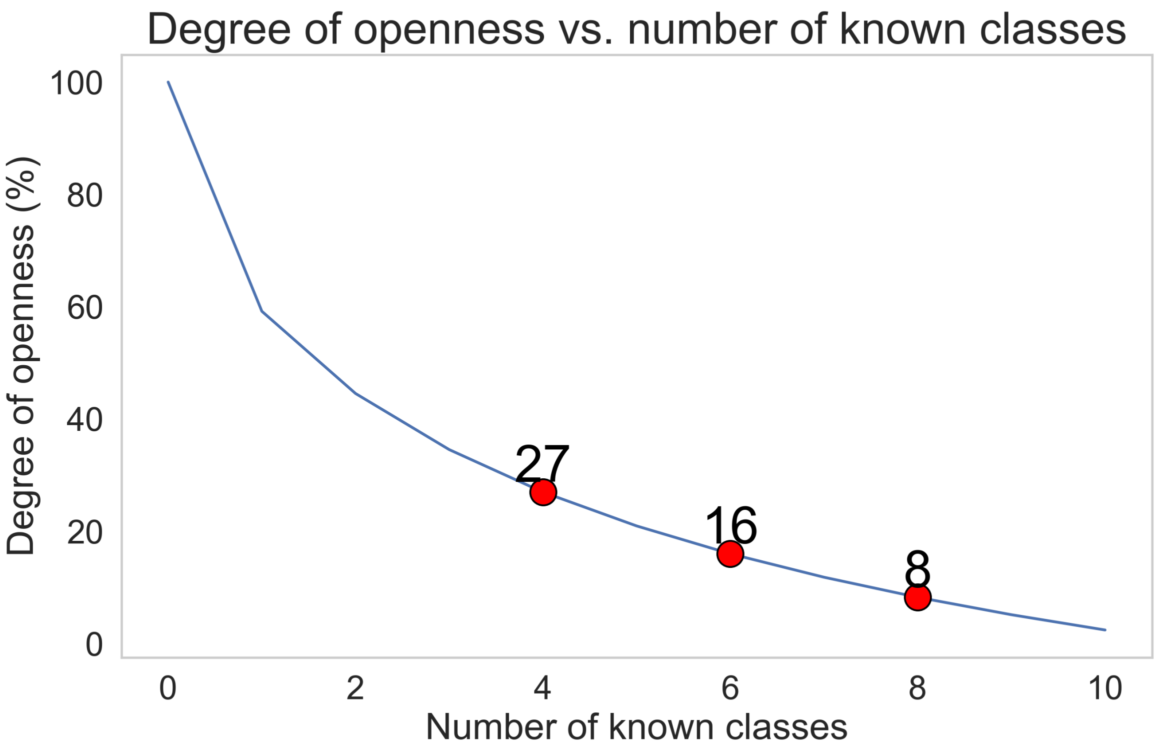


Figure 4.13: Relationship between the number of known classes and degree of openness.

of openness: 8%, 16%, and 27%, as depicted in the figure. In our experimentation using the Fashion-MNIST dataset, we maintain all ten classes during the testing phase ($C_{\text{test}} = 10$). The number of known classes in the training phase varies as 8, 6, and 4, while the remaining classes are treated as the unknown class to be recognized alongside the known classes during inference ($C_{\text{target}} = C_{\text{train}} + 1$). This setup results in openness variations of 8%, 16%, and 27%. A higher openness score indicates a greater number of classes considered as unknown. The evaluation involves assessing the overall F1 scores of different models, and the results are depicted in Figure 4.14. This figure illustrates how the F1-score changes across different degrees of openness for each individual model. We observe that as openness increases, the overall performance of all models decreases. The rate of this decline is minimal for $\mathcal{L}s$ and $\mathcal{L}s + OM$ methods that shows more stability across values of openness for these models. The other important finding here

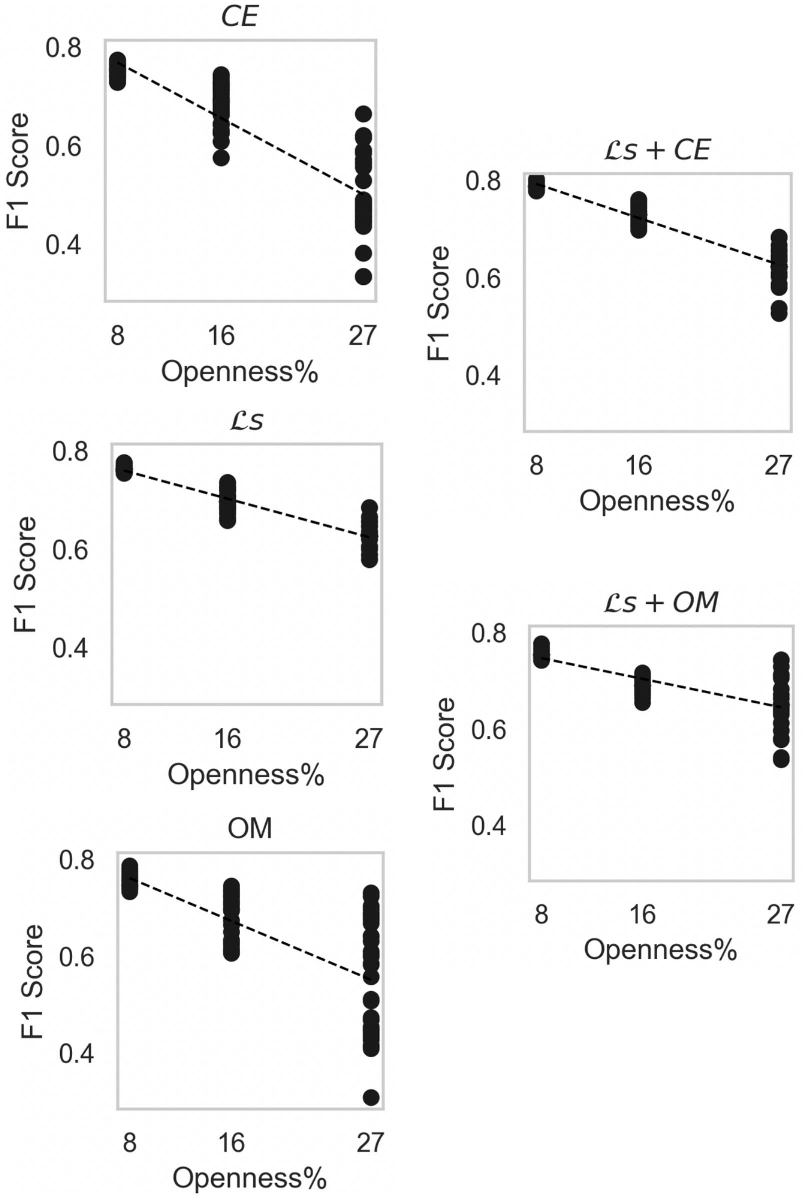


Figure 4.14: F1 scores against varying openness for Fashion-MNIST.

is that as we increase openness, the dispersion of the F1 score increases. However, when using loss functions $\mathcal{L}_s + CE$ and $\mathcal{L}_s + OM$, this dispersion reduces, indicating greater stability in these models. Therefore, as the number of unknowns grows, not only do we maintain a higher level of F1 score, but the variability among the 36 runs diminishes, demonstrating increased stability.

4.4.1 Summary

In this chapter, we assessed the performance metrics of the proposed superlative loss algorithm across three distinct datasets. We demonstrated that combining $\mathcal{L}_s$ with cross-entropy ($\mathcal{L}_s + CE$) and OpenMax ($\mathcal{L}_s + OM$) provides superior performance metrics compared to the standalone versions of CE and OM. Our evaluation comprised 36 runs for each model across the three different datasets. We showed that, in the majority of cases, the results exhibited statistically significant improvements, as indicated by p-values. Furthermore, the combined models of $\mathcal{L}_s + CE$ and $\mathcal{L}_s + OM$ exhibited increased performance and reduced variance with increasing openness compared to standalone models.

Chapter 5

Adversarial Examples

Research, such as [123, 63], has shown that DNNs are particularly vulnerable to fooling or adversarial examples, and the accuracy of these models can be reduced when facing these examples, as shown in [180]. Fooling examples are generated with the intention of being misclassified by the classifier as belonging to a particular class, often referred to as the target class. These examples are crafted in such a way that to human observers, they often appear as random noise or patterns. This means that the perturbations or alterations made to the input data are imperceptible or difficult for humans to discern. Despite being unrecognizable to humans, a DNN categorizes these examples with high certainty as members of the target class. This highlights the ability of adversarial examples to exploit the model's vulnerabilities, leading to incorrect classifications. Essentially, these artificially constructed examples are fully imperceptible to humans, but the classifier identifies them as members of the desired classes and labels them with high certainty (see Figure 5.1(c)). A more restrictive case is rejecting an adversarial example [180]– a visually similar input to the training dataset with small but intentional perturbations, such that it is mislabeled by a classifier as an entirely different class with high confidence (see Figure 5.1(d)).

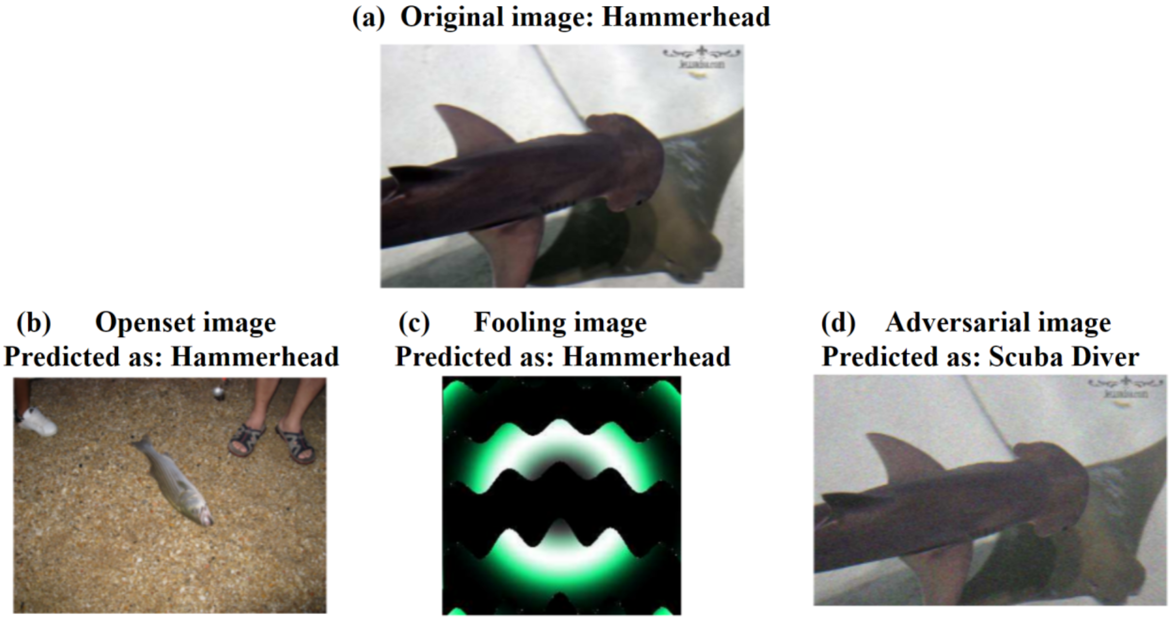


Figure 5.1: Examples of an original, open set, fooling, and adversarial images taken from [12]. (b) Example of a real image from an unknown category which is mapped to the class with the maximum response provided by Softmax. (c) A fooling input image which is unrecognizable to humans, but DNNs believe with high certainty to be a Hammerhead. (d) An adversarial image specifically constructed from hammerhead to scuba to fool DNNs into making an incorrect detection.

These inputs are derived from natural inputs in the training set and testing set. Suppose F is a trained DNN classification model and x is a natural input that is correctly classified, i.e., $C(x) = l$, where $C(x)$ means the classification of F on x predicted as class l. Then, an adversary can produce a new input x^* that is similar to x but is classified incorrectly, i.e., $C(x^*) \neq l$. In this simple case, the input is misclassified in a class different from the legitimate source class. A more restrictive case is where the adversary chooses a specific target class $t \neq l$ and crafts the adversarial example x^* close to x, and yet that gets misclassified such that $C(x^*) = t$. Adversarial example generation methods exploit gradient-based optimization for normal samples to find a small perturbation in a direction that maximizes the chance of misclassification. The radius of the search area around the natural samples and the used loss function to guide the direction are parameters that differ in these techniques.

[182] proposed the 'boundary tilting' perspective to explain the existence of adversarial samples for DNNs. They argued that adversarial examples stay near the classification boundaries and in regions where these boundaries are close to a sub-manifold of the data. [57] proposed that the true label of natural inputs can be changed by moving away from the manifold of the training data. Based on these studies, [50] assumed that by moving away from a point x belonging to a source class $C(x) = l$, the generated adversarial sample x^* must be pushed off of the data manifold and is classified incorrectly as $C(x^*) = \acute{l}$. Figure 5.2 depicts two sub-manifolds of circle and triangle that are separated by classification boundary (dashed line). Three possible situations for the location of adversarial examples are shown in a, b, and c insets, respectively. In these two-dimensional binary classification settings, x^* lies off its associated manifold.

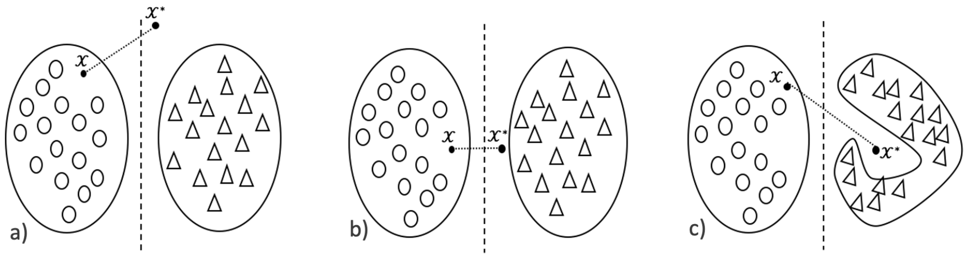


Figure 5.2: (a): The adversarial example x^* is generated by traversing away from '○' sub-manifold, but it is still far away from the '△' sub-manifold. (b): the adversarial example x^* is near the '△' sub-manifold but not on it, and it is also near the decision boundary (dashed line). (c): there exists a pocket in '△' sub-manifold that allows x^* lying in sub-manifold of the wrong label and far away from classification boundary.

This work [50] is based on performing kernel density estimation on the outputs of the final hidden layer learned by the model. Specifically, given the point x with predicted class l, the density estimate is defined as:

$$\hat{K}(x) = \frac{1}{|X_l|} \sum_{x_i \in X_l} exp(\frac{|Z(x) - Z(x_i)|^2}{\sigma^2}) \tag{5.1}$$

Where X_l is a set of training points with label l and $Z(x)$ is the output of the last hidden layer of a neural network for point x. The motivation behind using the outputs of the final layer is that this layer can provide more simplified manifolds to work with than input space. The deeper representations of a neural network can capture high-level semantic information about the input. Then, they investigate how far point x is from the final predicted class l. They argued that adversarial samples are likely to be in region of lower density estimates. Therefore, the classifier thresholds on the kernel density of the sample based on a selected threshold τ and reports x as adversarial if $\hat{K}(x) < \tau$.

While this approach can easily detect an adversarial example that is far from '△' sub-manifold (Figure 5.2.a), it does not work well for two cases: when x^* falls near '△' sub-manifold (Figure 5.2.b), and for a more difficult detection when x^* lies in the pocket of wrong sub-manifold (Figure 5.2.c). They showed that the density estimation of the adversarial sample decreases for the correct class and increases for the incorrect class. In the proposed feature space we aim to have a representation that pushes between-class distances and separates instances from different classes. In this setting different sub-manifolds are further apart and well separated which leads to larger spaces among them. Consequently, this approach has the capability to cover all the cases and the adversarial examples can easily be detected. We will discuss this capability in the next section.

For evaluation purpose, we use Fast Gradient Sign Method (FGSM)[63] to generate adversarial samples. The adversarial sample x^* in FGSM method is calculated as

$$x^* = x + \epsilon \, sign(\bigtriangledown_x J(\theta, x, y)) \tag{5.2}$$

In this respect, $J(\theta, x, y)$ denotes the model's loss function that specifies the cost of classifying the point x as label y. θ represents the model parameters and ϵ is a binary constant that controls the perturbation magnitude. This attack uses the derivative of the loss function with respect to the input feature vector. The original input x is perturbed in the direction of the loss gradient by magnitude ϵ. A FGSM is a type of white-box attack in which the attacker has access to the loss function and model parameters, in contrast to black-box attacks that the intruder does not have knowledge of the model.

The difficulty level of rejecting adversarial examples depends on how close the example is to the target class. For instance, if an adversarial example like a salmon shark is produced from a nearby class like a hammerhead, it will fail to be rejected as an unknown. However, if this example is generated from a faraway target class like scuba, it will be rejected as an unknown due to a remarkable difference in the output scores. That is why most of the studies proposed for OSR do not consider these examples in their experiments.

5.1 Experimental Results

In the experimental phase of our study, the CIFAR dataset was selected as the basis for our investigation. From the original set of 10 classes, we extracted four classes to serve as the unknown classes in our experiments. The test set comprised a combination of both known and unknown classes. Figure 5.3 illustrates a selection of samples from the CIFAR dataset, showcasing instances of known, adversarial, and unknown samples. In this context, the known classes include frog, cat, truck, airplane, deer, and bird. Adversarial examples are generated using the Fast Gradient Sign Method with an epsilon value of 0.1 applied to a batch of these known samples. The figure further includes open-set images representing classes such as horse, ship, dog, and automobile.

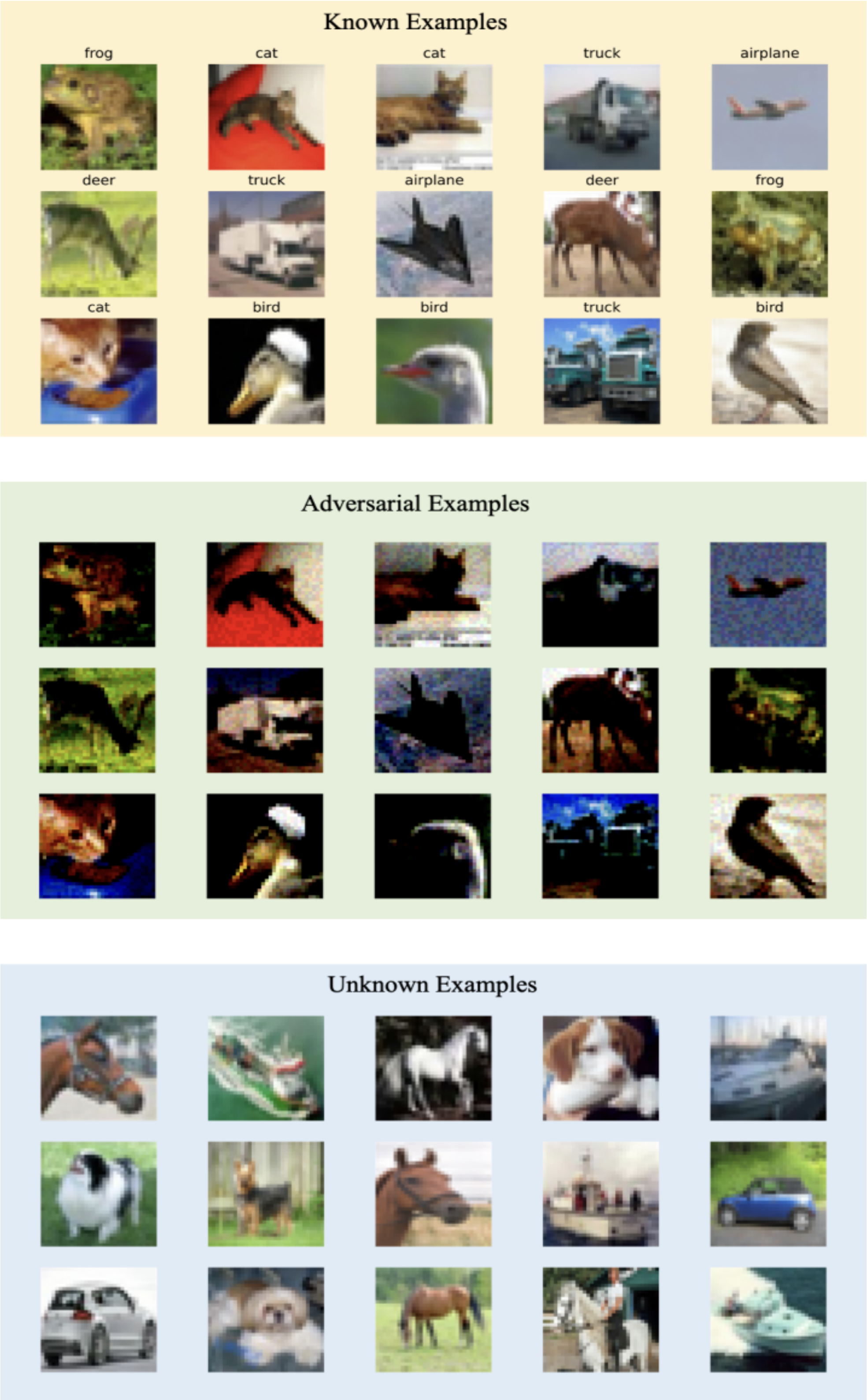
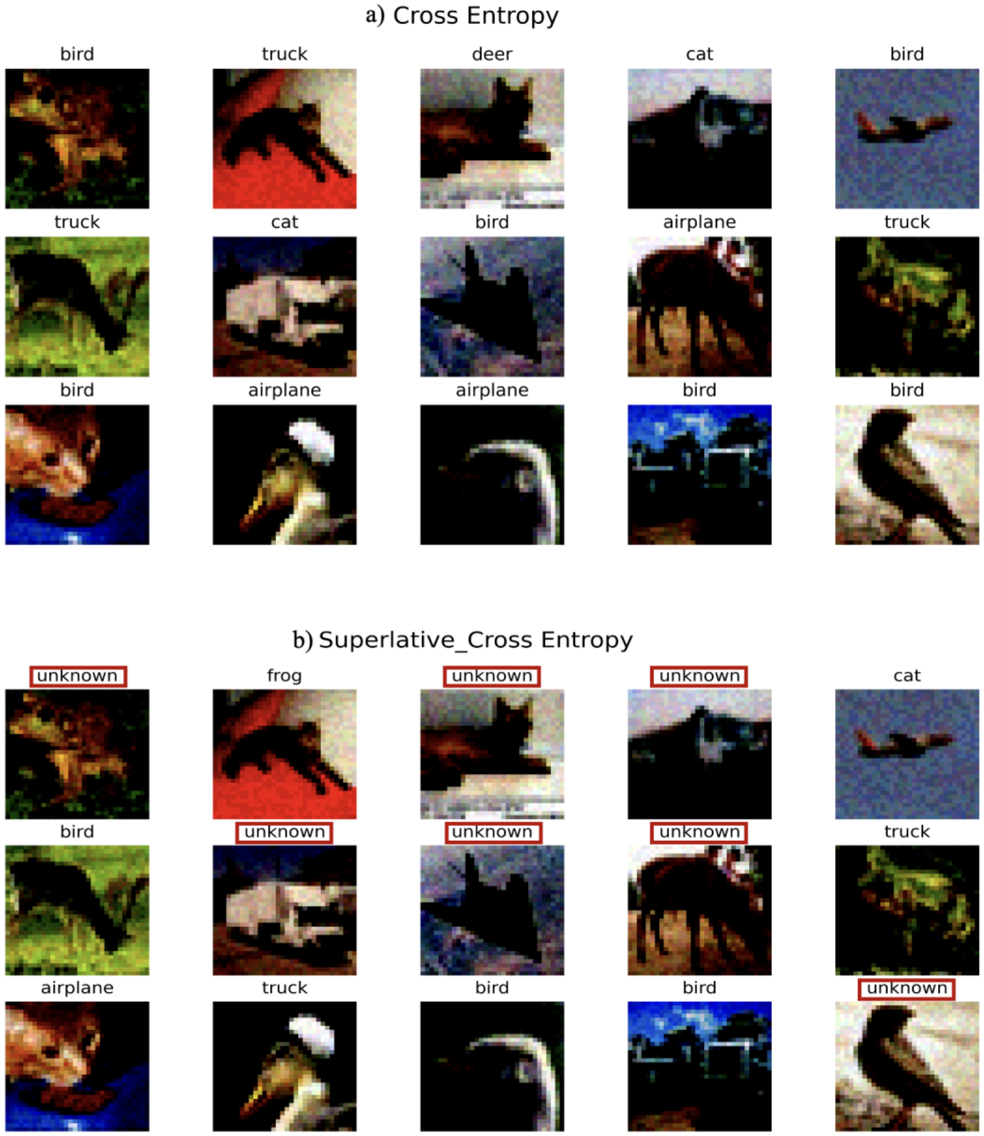


Figure 5.3: Examples of original, open set, and adversarial images for Cifar dataset.

Figure 5.4(a) depicts a conventional neural network using cross entropy, which notably struggles to identify adversarial examples. In this case, none of the adversarial examples can be detected. In contrast, Figure 5.4(b), where the superlative loss is combined with cross entropy, demonstrates a remarkable capability to detect 7 out of 15 adversarial instances as unknown, as evidenced by these 15 representative samples.



a) Cross Entropy

b) Superlative_Cross Entropy

Figure 5.4: Exploring the Cross-Entropy model for adversarial example detection and comparing its efficacy against a combined model approach.

Figure 5.5 further extends the exploration by showcasing the capability of the OpenMax model in detecting adversarial examples. Additionally, it highlights the effectiveness of superlative loss alone, as well as when combined with OpenMax. These models can automatically detect many unknown open-set and adversarial images, demonstrating an ability to classify adversarial examples as unknown.

To gauge the accuracy of these models in detecting adversarial samples, we conducted 12 distinct experiments for each of the five models. In each experiment, 10,000 random examples from the training dataset were selected, and adversarial examples were generated using Fast Gradient Sign Method with an epsilon of 0.1. The predicted labels for these examples were obtained, and using the true labels, the average accuracy was calculated over the 12 runs for each model. To calculate the adversarial example accuracy, the number of adversarial examples detected correctly as unknown is divided by the total number of adversarial examples. The comprehensive results of these experiments are presented in Table 5.1.

Table 5.1: Performance of Different Models on Adversarial and Unknown Accuracy

Methods	Adversarial Accuracy	Unknown Accuracy
Superlative_OpenMax ($\mathcal{L}s + OM$)	0.634392	0.556122
OpenMax (OM)	0.665556	0.561998
Superlative_Cross Entropy ($\mathcal{L}s + CE$)	0.821979	0.509434
Cross Entropy (CE)	0.527865	0.480681
Superlative ($\mathcal{L}s$)	0.752873	0.525353

5.1.1 Summary

In this chapter, we explore the significance of feature space exploration beyond open set recognition, particularly in the domain of adversarial example detection. We demonstrate that a conventional neural network employing vanilla cross entropy struggles to identify adversarial examples, as evidenced by its failure to detect any in our depicted scenario. However, upon integrating superlative loss with cross entropy, a notable improvement is observed in detecting 7 out of 15 adversarial examples as unknown. By showcasing the effectiveness of superlative loss both independently and in combination with OpenMax, we highlight the models' capacity to automatically detect numerous unknown open-set and adversarial images. This capability is instrumental in accurately detecting adversarial examples as unknown. In the upcoming chapter, we will consolidate our findings and conclusions, and outline potential avenues for future research.

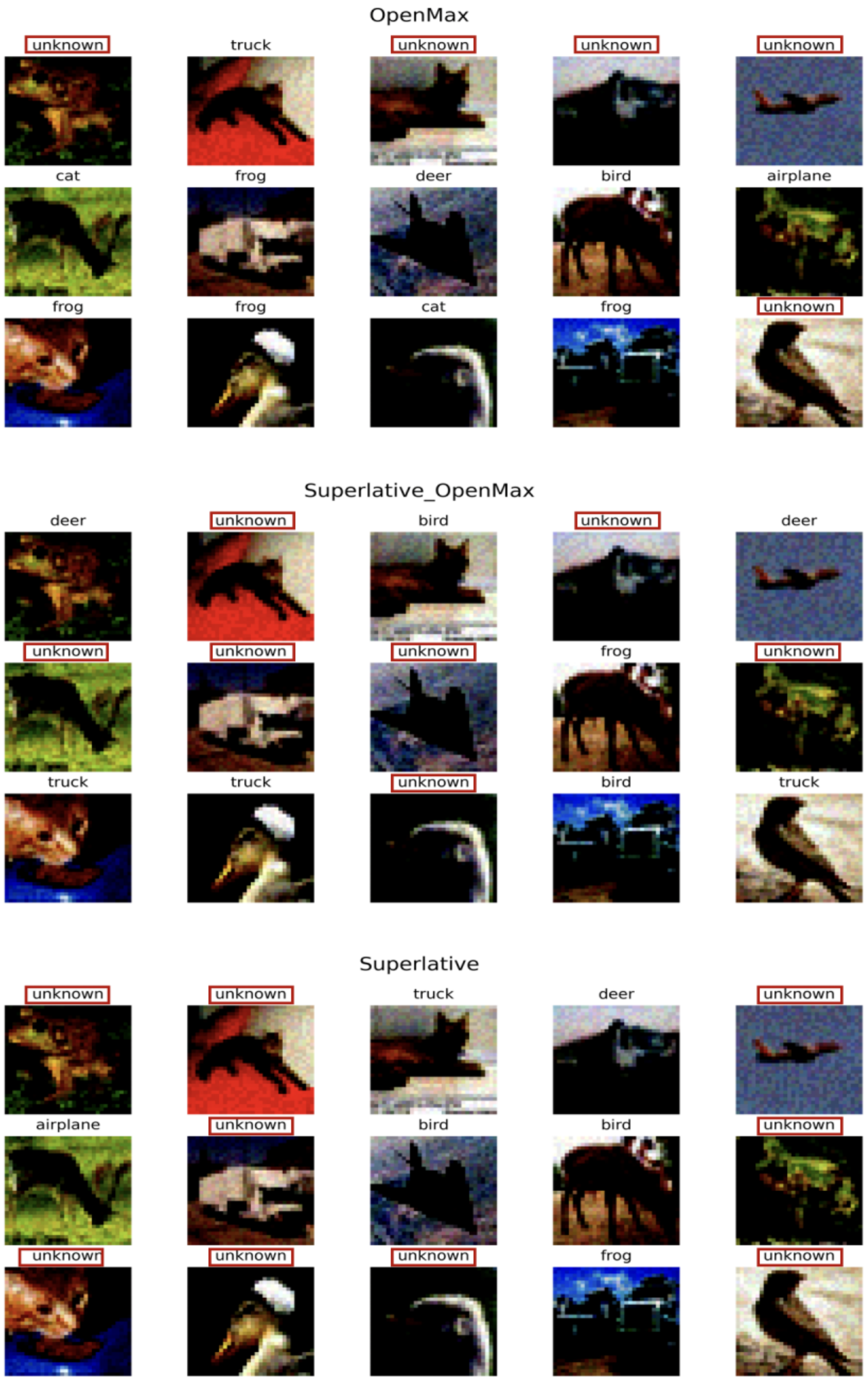


Figure 5.5: Exploring the OpenMax model for adversarial example detection and comparing its efficacy against Superlative and the combined model approaches.

Chapter 6

Conclusions

Machine learning-based techniques offer numerous opportunities and advancements for deriving deeper and more practical insights from data, which can be beneficial across various applications. However, the majority of these techniques primarily address the conventional closed-set scenario, where the label spaces for the training and test sets are identical. Open set recognition (OSR) seeks to address classification tasks in a manner that more closely resembles reality, focusing on not only classifying known classes but also effectively detecting unknown classes. In such scenarios, the training set may not encompass all possible classes, and the system must accurately identify unknown samples during testing.

However, constructing an accurate and comprehensive model in a dynamic real-world environment presents several challenges. It is often impractical to train for every potential example of unknown items, and models may fail when tested in diverse settings. This study introduces an algorithm that explores a novel representation of the feature space to enhance detection of unknowns in OSR tasks. The method aims to reposition and compact features to increase separation between samples from different classes while bringing samples from the same classes closer together, thereby enhancing the discriminative space and improving the detection of unknown samples.

The incorporation of Principal Component Analysis (PCA) into the optimization process proves highly advantageous in terms of simulation time, visualization, and performance. Particularly for datasets with a large number of features, the benefits of PCA are particularly evident. The performance of the proposed method is demonstrated on three established datasets, indicating its superiority over baseline methods in terms of accuracy and F1-score without requiring additional training time.

Categorized within the broader literature into three categories—training with borrowed additional data, training with generated additional data, and training without additional data—the proposed algorithm falls into the third category. Consequently, this robust model does not necessitate borrowing or generating additional data, nor does it require complex network architectures that can be both costly and time-consuming.

The proposed method is effective for addressing OSR problems in Deep Neural Networks and is adaptable to various loss functions and neural architectures. Its efficacy is demonstrated by combining it with both cross-entropy loss and OpenMax, a state-of-the-art algorithm. Furthermore, we illustrate that this new feature space is capable of effectively detecting adversarial examples. The parts of this dissertation were published in peer-reviewed conferences and are listed below:

Atefeh Mahdavi and Marco Carvalho. A survey on open set recognition. In 2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), pages 37–44. IEEE, 2021.

Atefeh Mahdavi and Marco Carvalho. Informed decision-making through ad- vancements in open set recognition and unknown sample detection. In 2024 57th Hawaii International Conference on System Sciences, pages 1090–1099, 2024.

Bibliography

[1] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. And: Autoregressive novelty detectors. *arXiv preprint arXiv:1807.01653*, 2018.

[2] Nicole Adler and Boaz Golany. Pca-dea: Reducing the curse of dimensionality. *Modeling data irregularities and structural complexities in data envelopment analysis*, pages 139–153, 2007.

[3] Charu C Aggarwal. Outlier analysis. In *Data mining*, pages 237–263. Springer, 2015.

[4] Igor Aleksander, Massimo De Gregorio, Felipe Maia Galvao França, Priscila Machado Vieira Lima, and Helen Morton. A brief introduction to weightless neural systems. In *ESANN*, pages 299–305. Citeseer, 2009.

[5] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[6] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Towards open-set identity preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6713–6722, 2018.

[7] Aleksander B Bapst, Jonathan Tran, Mark W Koch, Mary M Moya, and Robert Swahn. Open set recognition of aircraft in aerial imagery using synthetic template models. In *Automatic Target Recognition XXVII*, volume 10202, page 1020206. International Society for Optics and Photonics, 2017.

[8] Peter L Bartlett and Marten H Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(Aug):1823–1840, 2008.

[9] Daniele Battaglino, Ludovick Lepauloux, and Nicholas Evans. The open-set problem in acoustic scene classification. In *2016 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 1–5. IEEE, 2016.

[10] Brian Becker and Enrique Ortiz. Evaluating open-universe face identification on the web. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 904–911, 2013.

[11] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1893–1902, 2015.

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

[13] Lacey Best-Rowden, Hu Han, Charles Otto, Brendan F Klare, and Anil K Jain. Unconstrained face recognition: Identifying a person of interest from a media collection. *IEEE Transactions on Information Forensics and Security*, 9(12):2144–2157, 2014.

[14] Christopher M Bishop. Novelty detection and neural network validation. *IEE Proceedings-Vision, Image and Signal processing*, 141(4):217–222, 1994.

[15] Paul Bodesheim, Alexander Freytag, Erik Rodner, and Joachim Denzler. Local novelty detection in multi-class recognition problems. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 813–820. IEEE, 2015.

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

[17] TE Boult, S Cruz, A Dhamija, M Gunther, J Henrydoss, and W Scheirer. Learning and the unknown: Surveying steps toward open world recognition. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[18] Burak Cankaya, Kazim Topuz, and Aaron M Glassman. Business inferences and risk modeling with machine learning; the case of aviation incidents. *Business Inferences and Risk Modeling with Machine Learning; The Case of Aviation Incidents*, 11:1238, 2023.

[19] Douglas O Cardoso, Felipe França, and Joao Gama. A bounded neural network for open set recognition. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2015.

[20] Douglas O Cardoso, João Gama, and Felipe MG França. Weightless neural networks for open set recognition. *Machine Learning*, 106(9-10):1547–1567, 2017.

[21] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017.

[22] Enrique Castillo. *Extreme value theory in engineering.* Elsevier, 2012.

[23] Hakan Cevikalp. Best fitting hyperplanes for classification. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1076–1088, 2016.

[24] Hakan Cevikalp, Bedirhan Uzun, Okan Köpüklü, and Gurkan Ozturk. Deep compact polyhedral conic classifier for open and closed set recognition. *Pattern Recognition*, 119:108080, 2021.

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

[26] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[27] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *European Conference on Computer Vision*, pages 52–68. Springer, 2016.

[28] Chang-jun Chen, Yong-zhao Zhan, and Chuan-jun Wen. Hierarchical face recognition based on svdd and svm. In *2009 International Conference on Environmental Science and Information Application Technology*, volume 2, pages 692–695. IEEE, 2009.

[29] Zhilu Chen and Xinming Huang. End-to-end learning for lane keeping of self-driving cars. In *2017 IEEE intelligent vehicles symposium (IV)*, pages 1856–1860. IEEE, 2017.

[30] Giovani Chiachia, Alexandre X Falcao, Nicolas Pinto, Anderson Rocha, and David Cox. Learning person-specific representations from faces in the wild. *IEEE Transactions on Information Forensics and Security*, 9(12):2089–2099, 2014.

[31] C Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1):41–46, 1970.

[32] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[33] Filipe de O Costa, Ewerton Silva, Michael Eckmann, Walter J Scheirer, and Anderson Rocha. Open set source camera attribution and device linking. *Pattern Recognition Letters*, 39:92–101, 2014.

[34] Steve Cruz, Cora Coleman, Ethan M Rudd, and Terrance E Boult. Open set intrusion recognition for fine-grained attack categorization. In *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–6. IEEE, 2017.

[35] Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

[36] Milan Cvitkovic, Badal Singh, and Anima Anandkumar. Open vocabulary learning on source code with a graph-structured cache. *arXiv preprint arXiv:1810.08305*, 2018.

[37] Qing Da, Yang Yu, and Zhi-Hua Zhou. Learning with augmented class by exploiting unlabeled data. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[38] Rocco De Rosa, Thomas Mensink, and Barbara Caputo. Online open world recognition. *arXiv preprint arXiv:1604.02275*, 2016.

[39] Akshay Raj Dhamija, Manuel Günther, and Terrance Boult. Reducing network agnostophobia. In *Advances in Neural Information Processing Systems*, pages 9157–9168, 2018.

[40] Thomas G Dietterich. Steps toward robust artificial intelligence. *AI Magazine*, 38(3):3–24, 2017.

[41] Tri Doan and Jugal Kalita. Overcoming the challenge for text classification in the open world. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1–7. IEEE, 2017.

[42] Bernard Dubuisson and Mylene Masson. A statistical decision rule with incomplete knowledge about classes. *Pattern recognition*, 26(1):155–165, 1993.

[43] Hazım Kemal Ekenel, Lorant Szasz-Toth, and Rainer Stiefelhagen. Open-set face recognition-based visitor interface system. In *International Conference on Computer Vision Systems*, pages 43–52. Springer, 2009.

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

[45] Zhen Fang, Jie Lu, Feng Liu, Junyu Xuan, and Guangquan Zhang. Open set domain adaptation: Theoretical bound and algorithm. *arXiv preprint arXiv:1907.08375*, 2019.

[46] Jeanfranco D Farfan-Escobedo, Lauro Enciso-Rodas, and John E Vargas-Muñoz. Towards accurate building recognition using convolutional neural networks. In *2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, pages 1–4. IEEE, 2017.

[47] Geli Fei and Bing Liu. Social media text classification under negative covariate shift. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2347–2356, 2015.

[48] Geli Fei and Bing Liu. Breaking the closed world assumption in text classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 506–514, 2016.

[49] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[50] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[51] Barbel Finkenstadt and Holger Rootzén. *Extreme values in finance, telecommunications, and the environment.* CRC Press, 2003.

[52] Lydia Fischer, Barbara Hammer, and Heiko Wersing. Efficient rejection strategies for prototype-based classification. *Neurocomputing*, 169:334–342, 2015.

[53] Lydia Fischer, Barbara Hammer, and Heiko Wersing. Optimal local rejection for classifiers. *Neurocomputing*, 214:445–457, 2016.

[54] Yanwei Fu, Tao Xiang, Yu-Gang Jiang, Xiangyang Xue, Leonid Sigal, and Shaogang Gong. Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content. *IEEE Signal Processing Magazine*, 35(1):112–125, 2018.

[55] Giorgio Fumera and Fabio Roli. Support vector machines with embedded reject option. In *International Workshop on Support Vector Machines*, pages 68–82. Springer, 2002.

[56] Giorgio Fumera, Fabio Roli, and Giorgio Giacinto. Reject option with multiple thresholds. *Pattern recognition*, 33(12):2099–2101, 2000.

[57] Jacob R Gardner, Paul Upchurch, Matt J Kusner, Yixuan Li, Kilian Q Weinberger, Kavita Bala, and John E Hopcroft. Deep manifold traversal: Changing labels with convolutional features. *arXiv preprint arXiv:1511.06421*, 2015.

[58] ZongYuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. Generative openmax for multi-class open set classification. *arXiv preprint arXiv:1707.07418*, 2017.

[59] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in neural information processing systems*, pages 4878–4887, 2017.

[60] Chuanxing Geng and Songcan Chen. Collective decision for open set recognition. *arXiv preprint arXiv:1806.11258*, 2018.

[61] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, pages 9758–9769, 2018.

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

[63] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[64] R.L. Graham, D.E. Knuth, and O. Patashnik. *Concrete mathematics*. Addison-Wesley, Reading, MA, 1989.

[65] Yves Grandvalet, Alain Rakotomamonjy, Joseph Keshet, and Stéphane Canu. Support vector machines with a reject option. In *Advances in neural information processing systems*, pages 537–544, 2009.

[66] Edouard Grave, Moustapha M Cisse, and Armand Joulin. Unbounded cache model for online language modeling with open vocabulary. In *Advances in Neural Information Processing Systems*, pages 6042–6052, 2017.

[67] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

[68] Manuel Gunther, Steve Cruz, Ethan M Rudd, and Terrance E Boult. Toward open-set face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 71–80, 2017.

[69] Manuel Günther, Peiyun Hu, Christian Herrmann, Chi-Ho Chan, Min Jiang, Shufan Yang, Akshay Raj Dhamija, Deva Ramanan, Jürgen Beyerer, Josef Kittler, et al. Unconstrained face detection and open-set face recognition challenge. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 697–706. IEEE, 2017.

[70] Jingcai Guo, Han Wang, Yuanyuan Xu, Wenchao Xu, Yufeng Zhan, Yuxia Sun, and Song Guo. Multimodal dual-embedding networks for malware open-set recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[71] Blaise Hanczar and Michèle Sebag. Combination of one-class support vector machines for classification with reject option. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 547–562. Springer, 2014.

[72] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multi-view 3d object retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1945–1954, 2018.

[73] Dan Hendrycks, Mantas Mazeika, and Thomas G Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

[74] James Henrydoss, Steve Cruz, Ethan M Rudd, Terrance E Boult, et al. Incremental open set intrusion recognition using extreme value machine. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1089–1093. IEEE, 2017.

[75] Radu Herbei and Marten H Wegkamp. Classification with reject option. *Canadian Journal of Statistics*, 34(4):709–721, 2006.

[76] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[77] Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126, 2004.

[78] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

[79] Wladyslaw Homenda, Marcin Luckner, and Witold Pedrycz. Classification with rejection based on various svm techniques. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 3480–3487. IEEE, 2014.

[80] Tzu-Kuo Huang, Ruby C Weng, and Chih-Jen Lin. Generalized bradley-terry models and multi-class probability estimates. *Journal of Machine Learning Research*, 7(Jan):85–115, 2006.

[81] Andrei De Souza Inácio, Matheus Gutoski, André Eugênio Lazzaretti, and Heitor Silvério Lopes. Osvidcap: A framework for the simultaneous recognition and description of concurrent actions in videos in an open-set scenario. *IEEE Access*, 9:137029–137041, 2021.

[82] Anil K Jain and Stan Z Li. *Handbook of face recognition*. Springer, 2011.

[83] Lalit P Jain, Walter J Scheirer, and Terrance E Boult. Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision*, pages 393–409. Springer, 2014.

[84] A Jeya Christy and K Dhanalakshmi. Content-based image recognition and tagging by deep learning methods. *Wireless Personal Communications*, 123(1):813–838, 2022.

[85] Inhyuk Jo, Jungtaek Kim, Hyohyeong Kang, Yong-Deok Kim, and Seungjin Choi. Open set recognition by regularising classifier with fake data generated by generative adversarial networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2686–2690. IEEE, 2018.

[86] Felix Juefei-Xu and Marios Savvides. Multi-class fukunaga koontz discriminant analysis for enhanced face recognition. *Pattern Recognition*, 52:186–205, 2016.

[87] Pedro R Mendes Júnior, Roberto M de Souza, Rafael de O Werneck, Bernardo V Stein, Daniel V Pazinato, Waldir R de Almeida, Otávio AB Penatti, Ricardo da S Torres, and Anderson Rocha. Nearest neighbors distance ratio open-set classifier. *Machine Learning*, 106(3):359–386, 2017.

[88] Pedro Ribeiro Mendes Júnior, Terrance E Boult, Jacques Wainer, and Anderson Rocha. Specialized support vector machines for open-set recognition. *arXiv preprint arXiv:1606.03802*, 2016.

[89] Navid Kardan and Kenneth O Stanley. Mitigating fooling with competitive overcomplete output layer neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 518–525. IEEE, 2017.

[90] Shehroz S Khan and Michael G Madden. A survey of recent trends in one class classification. In *Irish conference on artificial intelligence and cognitive science*, pages 188–197. Springer, 2009.

[91] Zubair Ahmed Khan and Asma Rizvi. Ai based facial recognition technology and criminal justice: Issues and challenges. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(14):3384–3392, 2021.

[92] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[93] D.E. Knuth. Two notes on notation. *Amer. Math. Monthly*, 99:403–422, 1992.

[94] Sacha Krstulović. Audio event recognition in the smart home. In *Computational Analysis of Sound Scenes and Events*, pages 335–371. Springer, 2018.

[95] JT-Y Kwok. Moderating the outputs of support vector machine classifiers. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, volume 2, pages 943–948. IEEE, 1999.

[96] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2013.

[97] André Eugênio Lazzaretti, David Martinus Johannes Tax, Hugo Vieira Neto, and Vitor Hugo Ferreira. Novelty detection and multi-class classification in power distribution voltage waveforms. *Expert Systems with Applications*, 45:322–330, 2016.

[98] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.

[99] Fayin Li and Harry Wechsler. Open set face recognition using transduction. *IEEE transactions on pattern analysis and machine intelligence*, 27(11):1686–1697, 2005.

[100] Xue Li, Jinlong Fei, Jiangtao Xie, Ding Li, Heng Jiang, Ruonan Wang, and Zan Qi. Open set recognition for malware traffic via predictive uncertainty. *Electronics*, 12(2):323, 2023.

[101] Qing Lian, Wen Li, Lin Chen, and Lixin Duan. Known-class aware self-ensemble for open set domain adaptation. *arXiv preprint arXiv:1905.01068*, 2019.

[102] Bin Liang, Hongcheng Li, Miaoqiang Su, Xirong Li, Wenchang Shi, and Xiaofeng Wang. Detecting adversarial image examples in deep networks with adaptive noise reduction. *arXiv preprint arXiv:1705.08378*, 2017.

[103] Shengcai Liao, Zhen Lei, Dong Yi, and Stan Z Li. A benchmark study of large-scale unconstrained face recognition. In *IEEE international joint conference on biometrics*, pages 1–8. IEEE, 2014.

[104] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[105] Hong Liu, Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Qiang Yang. Separate to adapt: Open set domain adaptation via progressive separation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2927–2936, 2019.

[106] Jiachen Liu, Qiguang Miao, Yanan Sun, Jianfeng Song, and Yining Quan. Modular ensembles for one-class classification based on density analysis. *Neurocomputing*, 171:262–276, 2016.

[107] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[108] Zhun-ga Liu, Yi-min Fu, Quan Pan, and Zuo-wei Zhang. Orientational distribution learning with hierarchical spatial attention for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[109] Tong Luo, Kurt Kramer, Dmitry B Goldgof, Lawrence O Hall, Scott Samson, Andrew Remsen, and Thomas Hopkins. Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research*, 6(Apr):589–613, 2005.

[110] Atefeh Mahdavi and Marco Carvalho. A survey on open set recognition. In *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 37–44. IEEE, 2021.

[111] Atefeh Mahdavi and Marco Carvalho. Informed decision-making through advancements in open set recognition and unknown sample detection. *In 2024 57th Hawaii International Conference on System Sciences*, pages 1090–1099, 2024.

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

[113] Markos Markou and Sameer Singh. Novelty detection: a review—part 2:: neural network based approaches. *Signal processing*, 83(12):2499–2521, 2003.

[114] Stephen Marsland. Novelty detection in learning systems. *Neural computing surveys*, 3(2):157–195, 2003.

[115] Jane E Mason, Michael Shepherd, and Jack Duffy. An n-gram based approach to automatically identifying web page genre. In *2009 42nd Hawaii International Conference on System Sciences*, pages 1–10. IEEE, 2009.

[116] John McCarthy and Patrick J Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Readings in artificial intelligence*, pages 431–450. Elsevier, 1981.

[117] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013.

[118] Ali Moeini, Karim Faez, Hossein Moeini, and Armon Matthew Safai. Open-set face recognition across look-alike faces in real-world scenarios. *Image and Vision Computing*, 57:1–14, 2017.

[119] Russell Muzzolini, Yee-Hong Yang, and Roger Pierson. Classifier design with incomplete knowledge. *Pattern Recognition*, 31(4):345–369, 1998.

[120] Luiz C Navarro, Alexandre KW Navarro, Anderson Rocha, and Ricardo Dahab. Connecting the dots: Toward accountable machine-learning printer attribution methods. *Journal of Visual Communication and Image Representation*, 53:257–272, 2018.

[121] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 613–628, 2018.

[122] Manuel Alberto Córdova Neira, Pedro Ribeiro Mendes Júnior, Anderson Rocha, and Ricardo Da Silva Torres. Data-fusion techniques for open-set recognition problems. *IEEE Access*, 6:21242–21265, 2018.

[123] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[124] Enrique G Ortiz and Brian C Becker. Face recognition for web-scale datasets. *Computer Vision and Image Understanding*, 118:153–170, 2014.

[125] Poojan Oza and Vishal M Patel. C2ae: Class conditioned auto-encoder for open-set recognition. *arXiv preprint arXiv:1904.01198*, 2019.

[126] Poojan Oza and Vishal M Patel. Deep cnn-based multi-task learning for open-set recognition. *arXiv preprint arXiv:1903.03161*, 2019.

[127] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

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

[129] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[130] Jaewoo Park, Cheng Yaw Low, and Andrew Beng Jin Teoh. Divergent angular representation for open set image recognition. *IEEE Transactions on Image Processing*, 31:176–189, 2021.

[131] Jaewoo Park, Hojin Park, Eunju Jeong, and Andrew Beng Jin Teoh. Understanding open-set recognition by jacobian norm and inter-class separation. *Pattern Recognition*, 145:109942, 2024.

[132] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015.

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

[134] Pramuditha Perera and Vishal M Patel. Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, 2019.

[135] P Jonathon Phillips, Patrick Grother, and Ross Micheals. Evaluation methods in face recognition. In *Handbook of face recognition*, pages 551–574. Springer, 2011.

[136] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[137] Allan Pinto, William Robson Schwartz, Helio Pedrini, and Anderson de Rezende Rocha. Using visual rhythms for detecting video-based facial spoof attacks. *IEEE Transactions on Information Forensics and Security*, 10(5):1025–1038, 2015.

[138] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[139] Pierre Poitevin, Michel Pelletier, and Patrick Lamontagne. Challenges in detecting uas with radar. In *2017 International Carnahan Conference on Security Technology (ICCST)*, pages 1–6. IEEE, 2017.

[140] Sridhama Prakhya, Vinodini Venkataram, and Jugal Kalita. Open-set deep learning for text classification. *Machine Learning in Computer Vision and Natural Language Processing; ACM: New York, NY, USA*, pages 1–6, 2017.

[141] Sridhama Prakhya, Vinodini Venkataram, and Jugal Kalita. Open set text classification using cnns. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 466–475, 2017.

[142] Dimitrios Pritsos, Anderson Rocha, and Efstathios Stamatatos. Open-set web genre identification using distributional features and nearest neighbors distance ratio. In *European Conference on Information Retrieval*, pages 3–11. Springer, 2019.

[143] Dimitrios Pritsos and Efstathios Stamatatos. Open set evaluation of web genre identification. *Language Resources and Evaluation*, 52(4):949–968, 2018.

[144] Dimitrios A Pritsos and Efstathios Stamatatos. Open-set classification for automated genre identification. In *European Conference on Information Retrieval*, pages 207–217. Springer, 2013.

[145] Manikandasriram Srinivasan Ramanagopal, Cyrus Anderson, Ram Vasudevan, and Matthew Johnson-Roberson. Failing to learn: Autonomously identifying perception failures for self-driving cars. *IEEE Robotics and Automation Letters*, 3(4):3860–3867, 2018.

[146] Ajita Rattani, Walter J Scheirer, and Arun Ross. Open set fingerprint spoof detection across novel fabrication materials. *IEEE Transactions on Information Forensics and Security*, 10(11):2447–2460, 2015.

[147] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[148] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[149] Marko Ristin, Matthieu Guillaumin, Juergen Gall, and Luc Van Gool. Incremental learning of ncm forests for large-scale image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3654–3661, 2014.

[150] Gunter Ritter and María Teresa Gallegos. Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Pattern Recognition Letters*, 18(6):525–539, 1997.

[151] Anderson Rocha, Walter J Scheirer, Christopher W Forstall, Thiago Cavalcante, Antonio Theophilo, Bingyu Shen, Ariadne RB Carvalho, and Efstathios Stamatatos. Authorship attribution for social media forensics. *IEEE Transactions on Information Forensics and Security*, 12(1):5–33, 2016.

[152] Alina Roitberg, Chaoxiang Ma, Monica Haurilet, and Rainer Stiefelhagen. Open set driver activity recognition. In *Intelligent Vehicles Symposium (IV)*. IEEE, 2020.

[153] Jason D Roos and Arnab K Shaw. Probabilistic svm for open set automatic target recognition on high range resolution radar data. In *Automatic Target Recognition XXVII*, volume 10202, page 102020B. International Society for Optics and Photonics, 2017.

[154] Mark A Rosso. User-based identification of web genres. *Journal of the American Society for Information Science and Technology*, 59(7):1053–1072, 2008.

[155] Andras Rozsa, Manuel Günther, and Terrance E Boult. Adversarial robustness: Softmax versus openmax. *arXiv preprint arXiv:1708.01697*, 2017.

[156] Ethan M Rudd, Lalit P Jain, Walter J Scheirer, and Terrance E Boult. The extreme value machine. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):762–768, 2017.

[157] Ethan M Rudd, Andras Rozsa, Manuel Günther, and Terrance E Boult. A survey of stealth malware attacks, mitigation measures, and steps toward autonomous open world solutions. *IEEE Communications Surveys & Tutorials*, 19(2):1145–1172, 2016.

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

[159] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 153–168, 2018.

[160] Walter J Scheirer. Extreme value theory-based methods for visual recognition. *Synthesis Lectures on Computer Vision*, 7(1):1–131, 2017.

[161] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772, 2012.

[162] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2317–2324, 2014.

[163] Walter J Scheirer, Anderson Rocha, Ross J Micheals, and Terrance E Boult. Meta-recognition: The theory and practice of recognition score analysis. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1689–1695, 2011.

[164] Matthew Scherreik and Brian Rigling. Multi-class open set recognition for sar imagery. In *Automatic Target Recognition XXVI*, volume 9844, page 98440M. International Society for Optics and Photonics, 2016.

[165] Matthew D Scherreik and Brian D Rigling. Open set recognition for automatic target classification with rejection. *IEEE Transactions on Aerospace and Electronic Systems*, 52(2):632–642, 2016.

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

[167] Rudolf Schraml, Luca Debiasi, Cristof Kauba, and Andreas Uhl. On the feasibility of classification-based product package authentication. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2017.

[168] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

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

[170] Lei Shu, Hu Xu, and Bing Liu. Doc: Deep open classification of text documents. *arXiv preprint arXiv:1709.08716*, 2017.

[171] Lei Shu, Hu Xu, and Bing Liu. Unseen class discovery in open-world classification. *arXiv preprint arXiv:1801.05609*, 2018.

[172] Yu Shu, Yemin Shi, Yaowei Wang, Yixiong Zou, Qingsheng Yuan, and Yonghong Tian. Odn: Opening the deep network for open-set action recognition. In *2018 IEEE international conference on multimedia and expo (ICME)*, pages 1–6. IEEE, 2018.

[173] Daniel L Silver, Qiang Yang, and Lianghao Li. Lifelong machine learning systems: Beyond learning algorithms. In *2013 AAAI spring symposium series*, 2013.

[174] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[175] Johannes Stallkamp, Hazim K Ekenel, and Rainer Stiefelhagen. Video-based face recognition on real-world data. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[176] Wallace Stevens. Efficient uncertainty estimation for open-set object detection. *Epistemic Uncertainty Estimation for Object Detection in Open-Set Conditions*, page 91, 2021.

[177] Y Sun, L Ding, X Wang, and X Tang. Face recognition with very deep neural networks, 2015.

[178] Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, et al. The limits and potentials of deep learning for robotics. *The International Journal of Robotics Research*, 37(4-5):405–420, 2018.

[179] Niko Sünderhauf, Feras Dayoub, Sean McMahon, Ben Talbot, Ruth Schulz, Peter Corke, Gordon Wyeth, Ben Upcroft, and Michael Milford. Place categorization and semantic mapping on a mobile robot. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 5729–5736. IEEE, 2016.

[180] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[181] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*, pages 270–279. Springer, 2018.

[182] Thomas Tanay and Lewis Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016.

[183] Jingjing Tang, Yingjie Tian, and Xiaohui Liu. Lgnd: a new method for multiclass novelty detection. *Neural Computing and Applications*, 31(8):3339–3355, 2019.

[184] David MJ Tax and Robert PW Duin. Uniform object generation for optimizing one-class classifiers. *Journal of machine learning research*, 2(Dec):155–173, 2001.

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

[186] David MJ Tax and Robert PW Duin. Growing a multi-class classifier with a reject option. *Pattern Recognition Letters*, 29(10):1565–1570, 2008.

[187] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. A deeper look at dataset bias. In *Domain adaptation in computer vision applications*, pages 37–55. Springer, 2017.

[188] Rafael Vareto, Samira Silva, Filipe Costa, and William Robson Schwartz. Towards open-set face recognition using hashing functions. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 634–641. IEEE, 2017.

[189] Vinodini Molukuvan Venkataram. *Open Set Text Classification Using Neural Networks*. PhD thesis, University of Colorado Colorado Springs. Kraemer Family Library, 2018.

[190] Edoardo Vignotto and Sebastian Engelke. Extreme value theory for open set classification-gpd and gev classifiers. *arXiv preprint arXiv:1808.09902*, 2018.

[191] Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):13, 2019.

[192] Marten Wegkamp et al. Lasso type classifiers with a reject option. *Electronic Journal of Statistics*, 1:155–168, 2007.

[193] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):9, 2016.

[194] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.

[195] Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1511–1519, 2015.

[196] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[197] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4582–4591, 2017.

[198] Hu Xu, Bing Liu, Lei Shu, and P Yu. Open-world learning and application to product classification. In *The World Wide Web Conference*, pages 3413–3419, 2019.

[199] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, Qing Yang, and Cheng-Lin Liu. Convolutional prototype network for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2358–2370, 2020.

[200] Yang Yang, Chunping Hou, Yue Lang, Dai Guan, Danyang Huang, and Jinchen Xu. Open-set human activity recognition based on micro-doppler signatures. *Pattern Recognition*, 85:60–69, 2019.

[201] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4016–4025, 2019.

[202] Chong You, Daniel P Robinson, and René Vidal. Provable self-representation based outlier detection in a union of subspaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3395–3404, 2017.

[203] Yang Yu, Wei-Yang Qu, Nan Li, and Zimin Guo. Open-category classification by adversarial sample generation. *arXiv preprint arXiv:1705.08722*, 2017.

[204] Ming Yuan and Marten Wegkamp. Classification methods with reject option based on convex risk minimization. *Journal of Machine Learning Research*, 11(Jan):111–130, 2010.

[205] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 2019.

[206] Erik Zamora and Wen Yu. Novel autonomous navigation algorithms in dynamic and unknown environments. *Cybernetics and Systems*, 47(7):523–543, 2016.

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

[208] Hongjie Zhang, Ang Li, Xu Han, Zhaoming Chen, Yang Zhang, and Yanwen Guo. Improving open set domain adaptation using image-to-image translation. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1258–1263. IEEE, 2019.

[209] Rong Zhang and Dimitris N Metaxas. Ro-svm: Support vector machine with reject option for image categorization. In *BMVC*, pages 1209–1218. Citeseer, 2006.

[210] Shanshan Zhang, Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. Towards reaching human performance in pedestrian detection. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):973–986, 2017.

[211] Arthur Zimek and Peter Filzmoser. There and back again: Outlier detection between statistical reasoning and data mining algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(6):e1280, 2018.