MSc Thesis

Knowing the Unknown: Open-World Recognition for Biodiversity Datasets

Rajesh Gangireddy

April, 2023

UNIVERSITY OF TWENTE.

intel®

Knowing the Unknown: Open-World Recognition for Biodiversity Datasets

by

Rajesh Gangireddy

Submitted to the Faculty of Electrical Engineering, Mathematics and Computer Science in partial fulfilment of the requirements for the degree of
Master of Science

Submitted on : April 19, 2023

Supervisors:		*Graduation committee* :	
Dr. ir. L.E. Hogeweg	(Intel)	Prof. dr. P.J.M. Havinga	University of Twente
Dr. ir. L.J. Cornelissen	(Intel)	Dr. ir. J.W. Kamminga	University of Twente
Dr. ir. J.W. Kamminga	(UT)	Dr. ir. L.J. Spreeuwers	University of Twente
		Dr. ir. L.E. Hogeweg	Intel
		Dr. ir. L.J. Cornelissen	Intel

An electronic version of this thesis is available at https://essay.utwente.nl/.

UNIVERSITY OF TWENTE.

We can know only that we know nothing. And that is the highest degree of human wisdom.

- Leo Tolstoy, War and Peace

Acknowledgements

It is with great pleasure that I present to you my thesis on open-world recognition, a topic that has captivated me ever since I started with deep learning-based computer vision. This assignment has been carried out with the AI research team at Intel Benelux BV and with support from the Pervasive Systems group at the University of Twente.

As I reflect on my journey towards this thesis, I am filled with gratitude for the many people who have helped me along the way. Firstly, I would like to express my deepest gratitude to my supervisors at Intel - Laurens Hogeweg and Ludo Cornelissen for their guidance, expertise, and unwavering support throughout this project. Their insights, constructive feedback, and encouragement have been invaluable in shaping my thinking and refining this research. Next, I would like to extend a special thanks to my university supervisor, Jacob Kamminga, for his thoughtful questions and detailed feedback which have been instrumental in ensuring the high quality of this thesis.

Next, I would also like to thank the University of Twente for providing me with the opportunity to pursue my master's program and for awarding me a scholarship, without which this journey would not have been possible.

I would like to extend my heartfelt appreciation to my friends and family, who have stood by me and supported me through this journey.

Lastly, I would like to thank you, the reader, for taking the time to read this thesis. I hope that this thesis will provide you with new insights and directions in the fascinating field of open-world recognition.

I wish you a happy reading. All the best from me, the author.

Summary

"We don't see things as they are, we see them as we are". This statement rings true not just for humans but also for computer vision models. It goes without saying that, the world is a vast and mysterious place, teeming with countless unknowns. When deployed in real-world applications, computer vision models often encounter images that belong to categories that they have not been trained to recognise. When faced with such an image, these models naively classify it into one of the categories they have been trained on. Such computer vision models can be unreliable and pose significant risks in safety-critical applications. Therefore, a computer vision model must be 'aware' of the open world and be able to recognise and reject an image from an 'unknown' category. This is called open-world recognition.

In this research work, several existing research gaps in the domain of open-world recognition or out-of-distribution detection (OOD) are identified and addressed. In literature, most of the existing OOD methods have been evaluated on a limited set of balanced datasets which do not resemble real-world distributions. This often leads to methods appearing better than they truly are. To address this gap, eight retrofittable OOD detection methods (MSP [1], Energy [2], ODIN [3], GradNorm [4], PCA FRE [5], DkNN [6], SLOF [7] and a proposed method - EnWeDi) are evaluated on long-tailed and fine-grained datasets. These datasets pose significant challenges in the task of OOD detection as they consist of many categories with few examples for the methods to learn from and have subtle differences between 'known' and 'unknown' categories.

Furthermore, due to a lack of a standardised way to quantify difficulty in OOD detection, the accuracies reported in the literature cannot be directly compared between methods. To tackle this, the concept of using domain similarity scores between datasets to quantify the difficulty in OOD detection is introduced. This score is a representation of the least amount of work needed to move the images of one dataset to match the distribution of another dataset.

Additionally, limitations of existing state-of-the-art (SOTA) OOD detection methods were identified and the EnWeDi method that overcomes these limitations is proposed. Out of the 14 OOD detection scenarios tested in this work, the proposed method outperforms the existing methods in 5 cases and performs similarly to the highest accuracy-achieving method(s) in 7 cases. Depending on the dataset, when compared with the corresponding SOTA method, the EnWeDi method shows a maximum improvement of 2.9%, 1.3% and 12.7% in AUROC, true positive rate and false positive rate respectively.

The effect of stacking image feature embeddings extracted from intermediate layers of a convolutional neural network for the task of OOD detection, which was overlooked in the literature, is investigated in this work. While stacking led to an improvement in OOD detection AUROC by 6.8% on one dataset, it also led to a 13.5% decrement in the same on another dataset. The reasons for this are studied and 'AutoCrop' - a robust way of stacking feature embeddings to improve OOD detection in the later case is proposed.

Overall, this research work proposes several concepts, an OOD detection method and valuable insights into the domain of open-world recognition. Through these contributions, this research work aims at taking a step closer to the ultimate goal of making computer vision models more 'open-world ready' and for AI systems that use them to be reliable, trustable and safe.

Contents

List of Figures

List of Tables

List of Acronyms

AD anomaly detection. 6, 7, 13, 24, 38, 39, 47, 48, 53, 54, 70, 71, 72, 76, 85, 86

AUROC area under the receiver operating characteristic curve. iii, 45, 46, 48, 49, 50, 51, 52, 53, 55, 56, 58, 60, 61, 62, 63, 64, 66, 67, 70, 71, 81, 83, 88

BFHC best-fitting hyperplane classifier. 8

CNN convolutional neural network. 3, 4, 12, 19, 20, 22, 34, 41, 44, 47, 62, 63, 79, 80, 85, 87, 101

CPU central processing unit. 40, 47, 63

DNN deep neural network. 1, 7, 10, 23, 24

EMD earth mover's distance. 33, 34

EnWeDi Entropy weighted nearest neighbour's distance. iii, 18, 19, 31, 32, 42, 45, 48, 50, 51, 52, 53, 54, 55, 58, 59, 63, 71, 73, 74, 75, 76, 78, 88, 89

EVT extreme value theory. 8, 10

Faiss Facebook AI Similarity Search. 13, 26, 27, 41, 47

FPR false positive rate. iii, 5, 11, 45, 46, 47, 48, 51, 52, 54, 55, 56, 57, 69, 70, 72, 88

FRE feature reconstruction error. 13, 23, 24, 25, 42, 64, 71, 82, 84

GPU graphics processing unit. 40, 47, 63

ID in-distribution. 3, 5, 6, 7, 8, 11, 12, 13, 15, 16, 19, 20, 21, 22, 23, 26, 27, 33, 34, 40, 42, 45, 53, 59, 60, 64, 66, 74, 87, 88, 102, 103

KL divergence Kullback-Leibler divergence. 12, 15, 22

kNN k-nearest neighbour. 9, 10, 12, 13, 25, 26, 27, 28, 29, 30, 31, 32, 42, 44, 47, 59, 63, 69, 75, 76, 78, 82, 83, 84, 88, 103, 104

LDOF local distance-based outlier factor. 9, 29, 103

LOF local outlier factor. 9, 29, 103

LRD local reachability density. 103

MAV mean activation vector. 10, 12

MSP maximum softmax probability. 11, 19, 49, 102

NCM nearest class mean. 9, 10, 28

ND novelty detection. 6, 7

OD outlier detection. 6, 7

OOD out-of-distribution. vii, 1, 2, 3, 4, 5, 6, 7, 10, 11, 17, 18, 19, 20, 21, 22, 23, 33, 34, 45, 48, 53, 66, 87

OSR open-set recognition. 6, 14, 15

PCA principal component analysis. 13, 14, 23, 24, 25, 42, 60, 78, 83, 84

ROC receiver operating characteristic. 45, 51, 52, 53, 56

SLOF simplified local outlier factor. 9

SPM softmax probability margin. 102

SVM support vector machine. 8

TPR true positive rate. iii, 5, 45, 46, 47, 48, 51, 54, 55, 56, 57, 58, 69, 70, 71, 72, 88

Chapter 1

Introduction

The accuracy of deep neural networks (DNNs) for image classification tasks has increased rapidly in recent years, paralleling or even outperforming human performance. For instance, the highest accuracy achieved on the ImageNet dataset [8] is 91.0% [9]. Such high accuracy scores have led to an increase in confidence to use the DNNs for real-world applications. However, once launched in real-world applications, the classification models that have achieved 'high accuracy' on paper, might become unreliable due to the lack of knowledge of open and changing environments. Multiple works [10, 1, 11, 12] have shown that neural network-based classifiers can make high-confidence predictions for images, that they have never been trained to classify in the first place. This happens when the classifier is deployed without accounting for the uncertainty that would appear outside of its training classes [13]. This is called the closed-set assumption. However, this assumption does not hold for classification models targeting real-world applications.

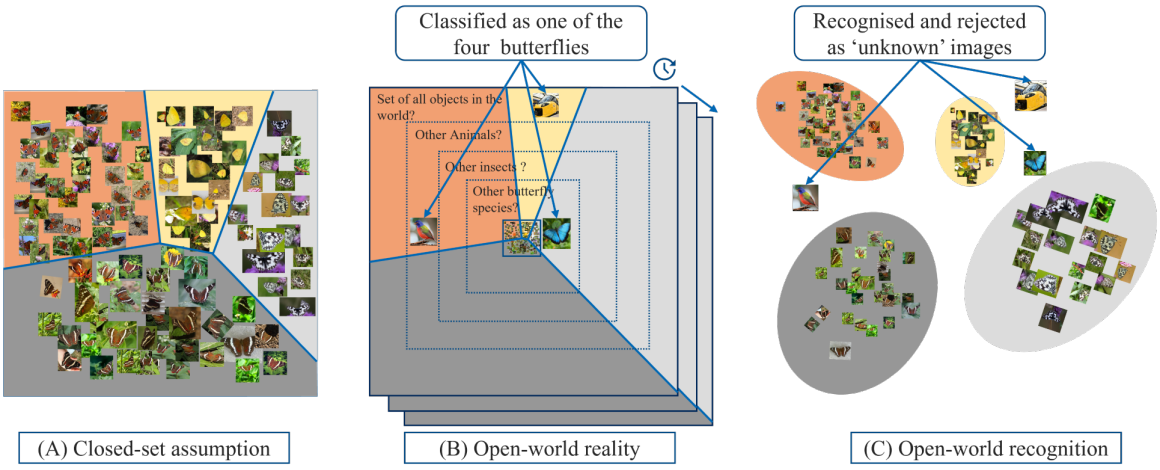


FIGURE 1.1: Example of a classifier trained on 4 species of butterfly. The closed-set assumption (A), the problem with the closed-set assumption (B) and the concept of open-world recognition (C).

To tackle this problem, techniques to make a classifier 'open-world ready' must be incorporated. Such techniques **make a classifier recognise the input images that do not belong to any of the classes it has been trained on and therefore 'reject' them**. This is referred to as open-world recognition and is more commonly known as **out-of-distribution (OOD) detection.**

OOD detection is essential not just for higher accuracies, but also for the reliability of

computer vision systems and to maintain safety in critical applications. For instance, in autonomous vehicles, when a vision system faces scenes or objects that it has not learnt to classify, it is better to 'reject' them as 'unknown' rather than misclassify them. In such cases, as the system cannot make a safe decision, it would be better to issue an alert or hand over control to humans.

A seemingly naive approach to the OOD detection problem is to extend the training set by adding a new class representing the unseen images that the classifier may encounter. However, the number of ways 'unknown' category objects can emerge is innumerable in an open world. In other words ***"All positive examples are alike; each negative example is negative in its own way"*** [14]. Hence, a very large number of such images are required to sufficiently generalise the 'unknown' category class. To implement this approach, an immensely large neural network would be required [3], making the training and the inference of the network impractical. This gives rise to the need for using OOD detection methods that can detect images from the 'unknown' category distribution rather than using a classifier that can classify images into the 'unknown' category class.

As OOD detection is increasingly used as a prerequisite for computer vision applications, there has been a growing interest in this domain in recent years. However, most of the existing research (Section 2.3) for OOD detection is benchmarked on a limited set of balanced datasets that do not fully capture the complex diversity and distribution of how objects occur in the real world. This makes the existing methods' applicability for practical applications debatable. This research aims to address this gap by evaluating OOD detection methods on datasets that resemble real-world characteristics.

Biodiversity datasets are often considered to closely resemble the distribution of objects in the real/open world because of the following reasons:

- **Long-tailed distributions:** Biodiversity datasets often tend to be long-tailed because of the inherent structure of natural ecosystems- a small number of species are very common, while a large number of species are rare or occur infrequently. This is similar to the distribution of objects in the real world, where there may be a few categories with a high frequency of occurrence and many categories with a low frequency of occurrence.

- **Fine-grained:** Biodiversity datasets are also often fine-grained, as they usually contain many closely related species. In the context of OOD detection, this property makes the task of classifying an image as 'unknown' or 'known' relatively difficult. Many real-world applications also face this challenge.

Furthermore, biodiversity datasets are subject to the same types of environmental and ecological changes that can impact other real-world datasets, such as extinction, a sudden increase in the number of species in an area or the discovery of a new species. These challenging conditions make biodiversity datasets an appropriate test case for OOD detection in real-world applications.

1.1 Research Gaps

From the extensive literature research (30+ research works presented in Section 2.3), the following gaps are identified for the task of out-of-distribution (OOD) detection:

Little to no research on OOD detection performance on long-tailed and fine–grained datasets Despite the considerable research in OOD detection, there is minimal

consensus on the best approach to tackle this problem for fine-grained and long-tailed datasets such as biodiversity datasets. Out of all the OOD detection methods mentioned in Section 2.3, only one method (OLTR [15]) has been tested on a long-tailed dataset.

No methodology to quantify OOD detection difficulty Most of the research works in the literature ignore quantifying the OOD detection difficulty. A few works quantify the OODness of a dataset into so-called near-, intermediate- and far-OOD subjectively. This could lead to OOD detection methods being tested on easy OOD datasets that might not resemble realistic, yet difficult-to-detect OOD images. This can result in an OOD detection method appearing better than it truly is.

Single OOD aspect solutions Almost all the OOD detection methods explored during the literature survey are evaluated on a single aspect of 'OODness' or OOD difficulty. However, in an open world, a classifier can face OOD images that have varying levels of OODness. Hence comprehensive testing of the methods on a range of OOD detection difficulties is required to make a classifier 'open-world ready'.

Not all methods can be retrofitted to an existing classifier A few methods that show promising results require re-training of the classifier and can not be retrofitted to an existing classifier. On a computer vision system, enabling OOD detection is more practical if it can be added without changing/retraining its classifier. This is mostly because training a model on a dataset can be time-consuming, especially if the model and/or dataset are large.

Limited use of feature embeddings from shallower layers For feature embedding-based methods, most of the methods extract feature embeddings from the penultimate layer of a CNN and do not experiment with shallower layers. Methods such as [5] report OOD performance on feature embeddings extracted from a few individual intermediate layers but not on feature embeddings stacked from multiple layers. A thorough search of the relevant literature yielded no relevant work that experimented with OOD detection methods on stacked feature embeddings.

1.2 Research Questions

To bridge the above-mentioned gaps in the domain of out-of-distribution (OOD) detection, this research aims to answer the following research question:

For in-distribution datasets, with fine-grained and long-tailed characteristics, what is the best retrofittable out-of-distribution (OOD) detection method that achieves high accuracy for the task of OOD detection across OOD datasets with varying difficulty?

This question can be further divided into the following sub-questions:

1. RQ1: *For a given in-distribution dataset, how can the difficulty of detecting OOD images from a particular OOD dataset be quantified?*
2. RQ2: *What improvements can be made to the existing OOD detection methods to further improve their accuracy for the task of OOD detection on fine-grained and long-tailed in-distribution datasets?*

3. RQ3: *What is the effect of stacking feature embeddings from the intermediate layers of a convolutional neural network on the accuracies of the OOD detection methods that make use of feature embeddings?*

4. RQ4: *What are the OOD detection methods that can consistently achieve the best accuracies over different OOD difficulties (from RQ1) when measured across a range of evaluation metrics?*

1.3 Thesis Layout

The thesis is organised as follows. Relevant background of the out-of-distribution (OOD) detection problem followed by a literature survey on OOD detection methods is presented in Chapter 2. The selected as well as the proposed OOD detection methods are described in Chapter 3. Chapter 4 describes the datasets, the experimental setup and the evaluation metrics on which the methods are evaluated. The results of these experiments are presented in Chapter 5. In Chapter 6, the insights from the results as well as recommendations based on these insights are presented. Finally, Chapter 7 concludes the thesis by answering the research questions as well as outlining potential areas of research for further exploration.

Chapter 2

Background

According to [16], open-world recognition consists of three steps:

1. Developing (or using an existing) classifier and detecting the images that are from the 'unknown' category.
2. Labelling the 'unknown' category images into new classes.
3. Using an incremental learning scheme to train the classifier on these new classes.

However, the term open-world recognition has been sparsely used in literature. Instead, the focus is on the first step mentioned above. Hence, **the term 'out-of-distribution detection' has been used more commonly to refer to the idea of open-world recognition.** Most of the existing research, including this research, considers them as synonyms.

In Section 2.1, the OOD detection task as a binary classification problem is formulated. A brief introduction to the problems related to OOD detection is presented in 2.2. A few existing concepts and methods for OOD detection are presented in Section 2.3. Finally, a few use cases of OOD detection are listed in Section 2.4.

2.1 Formulating OOD Detection Method

Out-of-distribution (OOD) detection is formulated as a binary classification problem where an input image is classified as in-distribution (ID) or out-of-distribution (OOD). The decision whether an image is ID or OOD is taken based on an OOD scoring function. The scoring function takes an image x and gives an OOD score for it which is compared with a preset threshold.

The decision of classifying an input image as ID or OOD can be represented as:

$$G_\lambda(x) = \begin{cases} ID & S(x) \geq \lambda, \\ OOD & S(x) < \lambda \ , \end{cases} \tag{2.1}$$

where $S(x)$ is the scoring function and λ is the threshold.

Setting the Threshold The threshold is determined using the validation set of ID dataset such that the desired true positive rate (or a false positive rate) on the ID images is reached. For instance, the threshold can be set to a value such that a large number (for instance 95%) of a set of known in-distribution images are classified correctly as ID. The set of validation images from the ID dataset can be used for this purpose.

The primary objective in out-of-distribution detection is to find a scoring function $S(x)$ that can maximise the accuracy on $G_\lambda(x)$ i.e., deciding whether an input image x belongs to ID or OOD.

2.2 Related Problems

All the related problems of out-of-distribution (OOD) detection can be formulated in terms of distribution shift. Distribution shift can be categorised into two types: covariate shift and semantic (label) shift. If X represents the input space and Y represents the label space, then a data distribution can be defined as a joint distribution of the probability $P(X;Y)$ in the space $X\text{x}Y$. A covariate shift is when the distribution of the data (input) changes during testing. In other words, covariate shift occurs when $P_{train}(Y|X) = P_{test}(Y|X)$ but $P_{train}(X) \neq P_{test}(X)$. In these shifts, the label space Y remains unchanged. On the other hand, in a semantic shift, the label space Y is different i.e., $P_{train}(Y) \neq P_{test}(Y)$.

Several problems that come under OOD detection in terms of motivation, as well as methodologies, are open-set recognition (OSR), anomaly detection (AD), novelty detection (ND), and outlier detection (OD). While most of the topics closely overlap with each other, there are a few differences among them in terms of the specific definition.

2.2.1 Open-Set Recognition

In open-set recognition, a classifier has two objectives: (1) classify images from 'known' classes with high accuracy and (2) detect images from the 'unknown' (or OOD classes).

Alternative taxonomy: OSR is also referred to as multi-class novelty detection, open category detection [17] and open set learning [18].

Relation to OOD detection: OSR is sometimes considered as an extension of OOD detection. OOD detection is similar to OSR in terms of objectives, motivation, and methodology. Multiple works use the terms interchangeably. However, there is a subtle difference between both. In terms of the scope of benchmarking and testing, in OSR, a single dataset is split into two subsets – the closed set (or in-distribution set) for training, and the out-of-distribution set for testing. A second dataset is not considered to be part of the original set and hence is not often considered in testing. In OOD detection, a 'set' is not defined and hence OOD data can be from an entirely different domain.

2.2.2 Anomaly Detection

Anomaly detection (AD) refers to the problem of finding patterns in data that do not conform to expected behaviour [19]. In other words, it is the detection of any anomalous samples that are deviated from the predefined normality during testing [20]. In a so-called sensory (input-related) AD, the 'normal data' is from an in-distribution dataset $P(X)$ and anomalies are from an out-of-distribution dataset $P'(X)$, where $P(x) \neq P'(X)$. However the labels remain unchanged ($P(Y) = P'(Y)$). On the other hand, in semantic anomaly detection, only semantic shift occurs i.e., ($P(Y) \neq P'(Y)$). In this work, only sensory anomaly detection is considered.

Relation to OOD detection: In the context of OOD detection, 'normality' corresponds to an in-distribution image and an 'anomaly' corresponds to an out-of-distribution image.

2.2.3 Novelty Detection

The task of novelty detection (ND) aims to detect any input images that do not fall into any training category. While novelty detection and anomaly detection are mostly used as synonymous, there is a fine difference between them. In novelty detection, the novel class images are not usually regarded as 'erroneous', or 'abnormal' as done in anomaly detection.

Relation to OOD detection: Novelty detection and OOD detection share the same objective. However, in novelty detection, the focus is more on detecting the semantic shift. Hence the novel images are treated as resources for improving the model, for instance, through active learning.

2.2.4 Outlier Detection

In outlier detection (OD), the objective is to detect images that appear to deviate significantly from the others in the given set of images[21].

Relation to OOD detection: In OOD detection, the in-distribution is defined during training, and the OOD images appear during testing time. In outlier detection, a single dataset containing both in-distribution and OOD (outlier) images are provided at the same time without any labels. In the case of outlier detection, the OOD is learnt during the training.

2.3 Literature Survey

In recent years, there has been a significant amount of research focused on the topic of out-of-distribution (OOD) detection and related problems, demonstrating a growing interest in this field. The goal of this literature survey chapter is to compile an overview of the existing OOD detection methods and related concepts, along with any of their noteworthy strengths and weaknesses. This overview will help to shortlist promising methods whose performance will be compared on biodiversity datasets. A few of the existing OOD detection methods from each category are reviewed in Subsections 2.3.2, 2.3.3, 2.3.4 and 2.3.5 respectively. Before starting an in-depth exploration of deep neural network (DNN)-based OOD detection methods, a few of the related solutions from the traditional machine learning (non-DNN) methods are studied. Most of the traditional machine learning-based (especially distance-based) OOD detection methods act as groundwork that is often adapted for DNN-based methods. A few such methods along with the OOD detection preliminaries introduced by them are discussed in the Subsection 2.3.1.

The existing set of OOD detection methods that use deep neural network can be categorised into:

- Output scores-based methods
- Feature embeddings-based methods
- Training-based methods
- OOD data generative methods

2.3.1 Concepts From the Traditional Machine Learning-based OOD Detection

Traditional classification and recognition tasks are based on leveraging the support vector machine (SVM) [22]. However, traditional SVM-based classifiers are trained with the assumption of a closed set where the entire classification space is open.

2.3.1.1 Concept of Bounded Classification Space

Open and closed classification spaces In [23], the authors propose the **1 vs. set machine**, which marks the spaces beyond a certain distance from the in-distribution as 'open space' and introduces the concept of risk of classifying an input belonging to this open space. As a solution, the authors propose creating a hyperplane parallel to the discriminating plane of the SVM to make a "slab". Every input that does not fall within this slab falls into the 'open space' and therefore is rejected. The authors in [24] extend the 1-vs-set method and propose a best-fitting hyperplane classifier (BFHC) that uses the kernel trick to make this method more practical to be applied for high dimensional feature spaces.

Quantifying open classification spaces In [25], the authors propose the solution of using a **compact abating probability (CAP) model** to further reduce the so-called 'open space risk'. In a CAP model, the probability of class membership reduces as the input sample moves from the in-distribution classes' space to the 'open space'. The first half of the CAP model is a one-class SVM model that calculates the probability of an input sample belonging to the 'closed space'. If the posterior estimate $P_o(y|x)$ of an input sample x predicted by this one-class SVM is less than a threshold, the input sample is called a 'negative' and is rejected outright. Otherwise, the input sample is fed to the next SVM which gives the posterior estimate $P_\eta(y|x)$. $P_\eta(y|x)$ gives the probability of an input belonging to one of the 'known classes' in the closed space. However, the main drawback is deciding on the threshold against which $P_o(y|x)$ is compared. The authors set this important threshold empirically and do not propose any methodology to calculate it.

Bounding the classification space of ID classes To further bound the quantified open spaces, the authors introduce the **probability-inclusion SVM** ($P_I - SVM$) in [26] which is based on the extreme value theory (EVT) [27]. The idea is that if there is sufficient knowledge of the in-distribution data, enough to generalise but not over-fit, then for each ID class, a decision boundary in the open space can be made. Each ID class then occupies a smaller, confined space in the complete feature space. Any input sample that does not fall within a threshold of these confined spaces can be treated as belonging to a 'novel class'. The major drawback of this method was the use of the same threshold at the decision boundaries for all the classes. This made it difficult to generalise effectively for imbalanced datasets. The **probabilistic open set SVM (POS-SVM)** classifier proposed in [28] overcomes this problem by setting a decision boundary threshold for each ID class instead of using a single threshold for all of them.

2.3.1.2 Concept of Using the Distance Between Samples for OOD Detection

Distance-based OOD detection concepts inspired by outlier detection methods are based on the nearest neighbour classifier. In these methods, for a given input sample, the nearest data points in a known set of training data points are searched. If the distance (or any

distance-based score) to these neighbours is above a threshold, the input sample can be classified as belonging to out-of-distribution.

In [16], the authors propose the **nearest non-outlier (NNO)** algorithm (based on the nearest class mean (NCM))classifier [29]). In this method, a distance between the test input and the mean of each known class is calculated. The test input is rejected if all these distances are beyond a certain threshold.

A slightly different method is presented in [30]. This method is an open-set version of the nearest neighbour classifier and is called **open-set nearest neighbour** (OSNN) method. Just like other methods, in the first step, the distance-based similarity score is calculated with every class. However, instead of comparing all the similarity scores calculated, OSNN calculates the ratio between the two highest similarity scores. If this ratio is below a threshold, then the input sample is considered to be belonging to OOD. This ratio is called the nearest neighbour distance ratio (NNDR) and is given by

$$NNDR = \frac{d(s, T_1)}{d(s, T_2)} \ , \tag{2.2}$$

where s is the input sample, T_1 and T_2 are the closest and the second closest neighbours of the sample s respectively. Here $d(x_1, x_2)$ indicates the Euclidean distance between sample x_1, x_2 in the feature space.The drawback of this method is its dependency on the accuracy of finding the neighbours T_1 and T_2. As the NNDR (and the whole method) depends on the distance of the input sample to the samples T_1 and T_2, T_1 and T_2 must not be outliers.

2.3.1.3 Concept of Using Density of Samples for OOD Detection

The density-based methods are more robust to noisy data points than distance-based methods. This is because these methods use local density estimates to detect outliers, rather than completely relying on global distance metrics which can be influenced by noisy or isolated examples in the training set used as reference.

One of the earliest density-based methods for detecting outliers is the **local outlier factor (LOF)** [31]. It is also one of the first methods in which an outlier is quantified on how outlying it is. For a given data point x, in LOF, the density of x is compared with the density of the k-nearest neighbours of x. A LOF value around 1 indicates that the data point x is located within a region of homogeneous density. Hence it can be assumed that the data point belongs to that local neighbourhood cluster of points. If the difference between the density in the local neighbourhood of x and the density around the kNNs of x is higher, x gets assigned a higher LOF value. The higher the LOF value of X, the more outlying x is considered to be. Although more stable and robust than the distance-based methods, LOF requires the calculation of a so-called reachability distance which is computationally expensive. For high-dimensional data points (such as feature embeddings of images), the complexity of LOF tends to be $O(n^2)$ [31].

Another density-based local outlier factor method called **simplified local outlier factor (SLOF)** reduces the computational complexity involved in LOF. For a given test point, the density of its neighbouring points is calculated using the distance to their respective k^{th} neighbours, instead of using the more computationally expensive reachability distance. However, for a single test data point, both LOF and SLOF require performing a kNN search at least k+1 times (once for the test point and once for every k nearest neighbour of the test point).

The **local distance-based outlier factor (LDOF)** [32] method is a density method which requires one kNN search for every test point. Given a datapoint x, LDOF uses the relative location of x to its neighbours to determine the degree to which it deviates from

its neighbourhood. For a data point x, LDOF is the ratio of 'the mean of the distances to its k-nearest neighbours' to 'the mean of the pairwise distances between those nearest neighbours'.

2.3.1.4 Concept of Using Reconstruction Error for OOD Detection

Sparse representation-based classifiers (SRC) identify the correct class by seeking the most sparsely distributed representation of the testing sample in terms of the training. In [33] the authors present an open-set version of the SRC. The method uses class reconstruction errors for classification. This means that it looks at how well a given input sample can be reconstructed using information extracted from the known classes of ID. Then the reconstruction error is used to classify the test sample as belonging to ID or OOD.

Another method is the **extreme value machine** [34] is based on the concept of margin data distribution. This method uses the statistical EVT and consists of two stages. In the first stage, the distributions of the matched reconstruction errors and the sum of non-matched reconstruction errors are modelled using the EVT. In the second stage, the reconstruction errors corresponding to a test sample from each class are calculated and the confidence scores based on the two distributions are fused to determine the identity of the test sample.

2.3.2 Output Scores-based Methods

The softmax function is often used as the output function for deep neural networks (DNNs) for multi-class classification. The output of the softmax function is the normalised probabilities for each class, all of which add up to 1. This normalisation makes DNNs with softmax as the output function, inherently of close-set nature.

OpenMax In [35], the authors introduce one of the first methods to adapt DNNs to an open-set setting by using OpenMax as a replacement for softmax. In addition to the probability of an input belonging to a known set of classes (ID), OpenMax also gives the probability of the input belonging to an unknown class (OOD) as output too. In this method, the activation vectors (scores from the penultimate layer) of all the training images are extracted. To these activation vectors, a nearest class mean (NCM) is applied to represent each of the known classes with a mean activation vector (MAV). For a test input image x, the activation vectors are extracted and additionally the top n classes that give a high probability for this input are noted. Next, the activation vector of the input and the MAV of the top n classes are fed to a softmax function (which considers an additional unknown class). This function gives the probabilities $P(Y_{C0,C1,C2..Cn}|x)$ where $C0$ represents the 'unknown class' and $C1...Cn$ are the top n classes. If $argmax(P(x)) = 0$ (0 is unknown class) or if $max(P(Y_{C0,C1,C2..Cn}|x) < threshold$, then the input x is classified as being from out-of-distribution. The authors use a grid search (rather than intuition) to find the threshold, the number of top classes to choose and other hyperparameters. These hyperparameters can be tuned to increase open-set image detection at the cost of rejecting more images belonging to true classes. However, the grid search requires a small set of calibration datasets consisting of OOD images to avoid extreme values for hyperparameters.

Maximum Softmax Probability A simple method is proposed in [1] to detect out-of-distribution inputs based on the observation that a well-trained neural network tends to assign higher softmax scores to ID examples than for out-of-distribution. The authors

use the maximum softmax probability (MSP) of input directly as the OOD score for that input. If this is below a threshold, the input is considered to be an OOD input. However, the assumption that softmax probability scores for OOD images are always well-spread across the classes leading to a lower MSP does not always hold true. The method is used as a common baseline in multiple OOD detection methods.

ODIN In [3], the authors propose ODIN (Out-of-DIstribution detector for Neural networks) method. This method uses a combination of temperature scaling [36] on the softmax score of the neural network along with the addition of small, controlled perturbations to inputs. This results in an increased gap between the softmax scores between in and out-of-distribution images. The input perturbations have a stronger effect on the images on which the neural network is trained (the in-distribution images) than on the out-of-distribution images. Although the method requires finding the best values for two hyperparameters - temperature scaling factor and input perturbation's magnitude, the results indicate that the method is quite effective in reducing the false positive rates (OOD image classified as ID image) when compared to the baseline method.

Generalised ODIN The Generalised ODIN [37] is an extension of the ODIN method [3] and combines decomposed probabilities with a modified input prepossessing to learn the out-of-distribution. There are essentially two probabilities that are learnt: the probability that the input is in-distribution ($p(d_{in}|x)$ and the probability that an input belongs to a class y given that the input is in-distribution ($p(y, d_{in}|x)$). These two probabilities are then used to calculate the probability

$$p(y|d_{in}, x) = \frac{p(y, d_{in}|x)}{p(d_{in}|x)} \ . \tag{2.3}$$

The two decomposed probabilities are less overconfident and can reject the OOD images better than a standard classifier.

Energy-based OOD detection method The authors in [2] present an energy-based out-of-distribution (OOD) detection method and use "energy scores" instead of softmax scores to detect OOD images. Energy-based models (EBM) [38] learn a function that maps each image from the input space to a single scalar value called the 'energy'. A large number of such mappings can be used to make a probability distribution function for the inputs. The idea behind this method is that the energy value of an OOD image would be further away from the centre of the probability density function of the ID images' energy. The authors present a new energy scoring function, based on the Helmholtz free energy function, that can be used as a scoring function for any pre-trained model. Moreover, the same energy function can be used as a trainable cost function, instead of softmax, to further improve the OOD detection performance. However, the authors compare the method with a few old baseline methods like [3] and [39] and not with the then-existing state-of-the-art OOD detection methods.

Rectified activations method Based on the observation that the internal activations of neural networks have highly distinctive signature patterns for OOD inputs, the authors in [40] propose the Rectified Activations (ReAct) method. The method can be used as such for OOD detection or can be applied to existing OOD detection methods to increase accuracy. The method shows that the mean activation from the penultimate layer for in-distribution data is mostly characterised by near-constant mean and standard deviation.

In contrast, for OOD data, the mean activation has a larger variation across the activation units. Rectifying (trimming) the activations at an upper limit leads to a better separation of the distribution curves of the vectors produced by ID and OOD data. The authors show that combining this method with the energy-based OOD detection method [2] further increases the performance.

GradNorm While most of the methods in this section estimate OOD scores using by directly using the output scores, the GradNorm [4] method uses the vector norm of gradients extracted from a trained neural network as an OOD scoring function. Gradients are back propagated from the Kullback-Leibler divergence (KL divergence) [41] between the softmax output and a uniform distribution. For an ID image, the prediction scores from the model tend to concentrate on one of the ground-truth classes and are therefore less uniformly distributed. The idea behind GradNorm is that the gradient norm of the Kullback-Leibler divergence will be higher for the in-distribution images than that for OOD images. This information is leveraged to calculate an uncertainty score to detect OOD images.

Competitive overcomplete output layer In [42], the authors propose the competitive overcomplete output layer (COOL) neural network as a way to reduce the overgeneralisation of neural networks over regions far from the training data. In this method, more than one output is allocated to each class to encourage the outputs to compete with each other during the process of stochastic gradient descent. This in turn results in partitioning the input space making a narrower area for each class around the training data distribution.

2.3.3 Feature Embeddings-based Methods

A feature embedding of an image extracted from a convolutional neural network (CNN) is the representation of different aspects of that image as a numerical vector of a fixed length. In general, the feature embeddings are extracted from the last (closest to the output) convolutional layer of the CNNs. A few methods that use feature embeddings rather than output scores for the task of OOD detection are mentioned in this section.

Deep k-Nearest Neighbours The authors in [43] introduce the deep k-nearest neighbours method which performs kNN search on the input image's features extracted at multiple intermediate layers of the CNN. Using this information, and the distance of the input image to the centroids (mean activation vector) of ID classes at each layer, a so-called credibility score is calculated. For an input image, any classification performed by the classifier must be supported by this credibility score. A very low credit score would indicate that the input image can not be trusted to belong to any of the ID classes. As an added advantage, the method can be used to answer why a certain image is classified into a certain class. This interpretability can be achieved by finding which training images were the nearest neighbours of the input image at each layer where kNN is performed. The method is also effective in detecting adversarial images [10]. One of the drawbacks of this method is that although the centroids for each ID class can be calculated offline, it could be computationally expensive to calculate kNN at every layer (especially with deep networks).

Distance to the k-nearest neighbours Using the k-nearest neighbours concept, [6] presents 'Out-of-Distribution Detection with Deep Nearest Neighbours'. The method makes use of the normalised feature embeddings extracted from the penultimate layer of the CNN. For an input image x, the method computes the Euclidean distance between the normalised feature vector of x to all the normalised feature embeddings of the ID images. The distances are sorted in ascending order to get the k-nearest neighbours. If the distance to the k^{th} neighbour is more than a predetermined threshold, the image is considered an OOD image. The threshold is fixed to a value such that a high fraction of in-distribution data is correctly classified. The authors also indicate that using the method with contrastive learning leads to a significant improvement in OOD detection performance. The method is relatively new (published in June 2022) and utilises Facebook AI Similarity Search (Faiss) library [44] for performing faster and more efficient kNN search.

Note on kNN-based methods While kNN has been used in many machine learning applications, it has largely been overlooked for applications where the number of data points and the number of features required to represent each data point is large. This is mainly due to the computational and memory requirements for performing kNN. As the distance from the test data points to all the training data points is calculated, kNN requires a lot of computational resources. With large datasets, the time and memory requirements can be prohibitive. (This is more crucial when distance metrics like Minkowski distance with p =2 (Euclidean) and above are used, as they require finding the p^{th} root). However, in the last few years, multiple fast approximate methods have been developed for performing kNN search, offering significant improvements in terms of speed and memory required.

Classification-Reconstruction learning Classification-Reconstruction learning algorithm for open-set recognition (CROSR) [45] is a method that aims in learning feature representations that can be used to detect OOD images while also classifying images within ID classes accurately. The open-set classifier consists of two parts: a closed-set classifier and an 'unknown class' detector. Both parts utilise a deep classification-reconstruction network. The known-class classifier leverages the supervised learned predictions and the 'unknown class' detector uses a reconstructive latent representation. This enables the unknown class detector to make use of a wider pool of features that may not be discriminative for known classes. However, the authors' implemented the method in the less used Chainer Framework [46] making it less attractive to be used in this work.

Feature Reconstruction Error Using PCA The authors in [5] propose a fast feature reconstruction error (FRE)-based method for OOD and anomaly detection. This method is based on modelling the sub-space of the feature embeddings extracted from a CNN by applying linear principal component analysis (PCA) on them. In this approach, a set of ID images' feature embeddings are extracted from an intermediate (usually the penultimate) layer of a CNN. A PCA model learns to reduce these feature embeddings into feature vectors of a lower dimension as well as reconstruct those reduced feature vectors back to their original dimensions. During the inference (OOD detection) phase, the input image's feature embedding is reduced and reconstructed using this PCA model and a FRE is calculated. As the PCA model has been learnt the reduction and reconstruction transformations only using ID images, a higher FRE would mean that the input image belongs to OOD. This FRE approach can be followed either by training a single PCA model on the whole ID dataset or using a PCA model for each class in the ID dataset.

Furthermore, the authors also experiment with this approach with non-linear methods like kernel PCA.

2.3.4 Training-based Methods

Minimum Others Score The method in [47] introduces a new scoring function called minimum others score (MOS). The idea behind this method is to decompose the large semantic space into smaller groups of subspaces which represent similar concepts of an image (for instance, all the visually similar Fungi classes could be grouped into a group, while all the butterflies into another). This group-based learning is done as a training process. For each group, apart from the individual classes in it, an 'others' class is also added. During the forward pass, a confidence score that the image belongs to a certain class of the subgroup as well as to this "others" class in the subgroup is calculated using MOS. If an input image gets a high confidence score (greater than a threshold) to be in the "others" class in all the groups, the image is predicted to be an OOD image.

tWiSARD In [48], the authors propose a method using WiSARD [49], a weightless neural network model. In WiSARD, for a given input, a fitness score indicating how well it fits into each class is calculated. The input is assigned to the class for which it received the highest fitness score. Using this WiSARD model, the similarity (or fitness score) between the observations from training data and the known classes (in ID) is calculated to define the boundaries between each known class. During the test time, the input's fitness score is then used to estimate the probability of it not belonging to any of the known classes.

CIDER In [50], the authors propose CIDER, a Compactness and DispErsion Regularized learning framework for OOD detection. The objective is to make different classes relatively far apart while making the samples within each class form a compact cluster. This is done in a hyperspherical space with the goal of having a high angular distance between clusters of classes. The authors combine a new dispersion loss (for inter-class dispersion) and a compactness loss (for intra-class compactness) during the training phase to achieve this.

Class Conditioned Auto-Encoder The class conditioned auto-encoder for open-set recognition (C2AE) [51] method uses auto-encoders and a new training procedure split into two sub-tasks: closed-set classification and open-set identification. The first half of the method is an auto-encoder and which feeds its output to a classifier. During the first phase of training, both the encoder and the classifier are trained to reduce the loss on multi-class classification ID images. In the second stage of training, the encoder with fixed weights and the decoder are trained to perfectly and poorly reconstruct the feature embeddings of ID and OOD images respectively. During the inference stage, apart from the classification probabilities by the classifier, a reconstruction error is calculated for each image using the encoder-decoder pair. If this reconstruction error is higher than a set threshold, the image is considered to be from OOD.

Outlier Exposure In [39], the authors introduce an Outlier Exposure method which in a simple and effective way can improve existing OOD detection methods. This is achieved by learning effective heuristics for detecting OOD inputs by exposing the model to OOD examples. This, supposedly, makes the model more aware of the inliers, thus learning a more conservative concept of the inliers and hence being able to discriminate the OOD inputs. However, in order to select these outlier examples, it is required to estimate the

distribution from which OOD images can appear during the test phase. This is not always practical.

Self-Supervised Learning with Auxiliary Rotation Loss In [52] the authors train a classifier with an auxiliary self-supervised rotation loss for OOD Detection. The method is based on the observation that the maximum softmax probability $max_c P(y = c|x)$ for an input x is higher for in-distribution (ID) input than for OOD input. Post training with the loss, the images from the ID test set and the OOD set are scored using the Kullback-Leibler divergence [41]. These scores can be compared with a threshold to classify the images as ID or OOD. Although the performance of the methods is similar to existing methods, the authors suggest that adding the self-supervised auxiliary rotation loss to existing OOD detection methods can further increase their performances.

Posterior sampling-based outlier mining The posterior sampling-based outlier mining (POEM) framework for OOD detection is introduced in [53]. POEM selects the most informative outlier data during the training which can help a classifier estimate a decision boundary between ID and OOD. The subset of outliers is selected from the auxiliary outlier set using Thompson sampling in an action-reward optimisation process to build this decision boundary. Although the results are comparable with SOTA methods, the requirement of such a so-called auxiliary outlier set is the main drawback of the method.

Open Long-Tailed Recognition The open long-tailed recognition (OLTR) [15] method deals with open-set recognition in long-tailed and open-ended distributions. This training method handles imbalanced classification, few-shot learning, and open-set recognition. It uses two concepts to do this: dynamic meta embeddings and modulated attention, for transfer learning between head and tail and discrimination between head and tail respectively. The dynamic meta embedding, in turn, is made up of two features: the direct feature from the input image and a memory feature inspired by meta-learning. The direct feature is used to keep the centroids for each class as far as possible. The memory feature captures visual concepts from training classes, retrieved from memory with a much shallower model. For differentiating the 'head classes' from the 'tail classes' as well as the open set classes, the concept of modulated attention [54] is adapted. Although the method is complicated and requires 2 stages of training the classifier, the method is the most promising for datasets having a long-tailed distribution.

2.3.5 Generative Methods

In the methods discussed above, while most of the methods use only the knowledge from ID images, a few methods need examples of OOD images to estimate the ID boundaries. In most of the OOD detection tasks, a few OOD examples can never be enough to generalise the out-of-distribution. In this section, the methods that try to overcome this by generating OOD images for approximating the OOD are presented.

Generative OpenMax The authors, in [55], present the Generative OpenMAX (G-OpenMAX) method which extends the work [35] through novel class image synthesis using GANs. The idea is that by generating the OOD images, the decision boundary between ID images and OOD images can be better estimated using the knowledge of both distributions. This leads to a better balance between open space and closed classification space. The OOD images are generated using a Generative and Adversarial Network (GAN) [56] and

are grouped into a new class. The classifier is trained with a set of known classes (ID images) along with an additional 'unknown class' consisting of OOD images. The rest of the method is similar to the OpenMAX method [35].

The work lays out multiple fundamental concepts for artificially generating images for the unknown class. The images generated have to be highly distinct compared to the images from the known classes while being general enough to represent as much open space as possible. The authors also suggest that mixing parts (or features) of images from multiple known classes is a good approximation to generate an image of an 'unknown class'. Doing so will result in an image that is still likely in the same domain and still does not belong to any of the originally known classes (of ID). Since all known classes in a given dataset are encoded by one-hot vectors, a new class image can be generated by (and represented through) a linear combination of these known classes in latent space. However, additional filtering is required to filter out the generated images that are similar to images of a known class. Hence, these generated images are inferred on another classifier that has been trained already on known classes, and only the incorrectly predicted images are selected as candidates for the k+1 class for training.

There are multiple limitations of this method. First, the methods assume only a tiny subspace of the open set by generating unknown class images using known classes. Second, the authors tested the method only on simple monochrome datasets containing digits and alphabets (MNIST [57] and HASYv2 [58]). More importantly, the authors mention that the generation of unknown class images through this method needs at least 500 images per class which is not always possible, especially in a biodiversity dataset.

Open Set Learning with Counterfactual Images In [59], the authors propose a new data augmentation technique, called counterfactual image generation for open-set recognition (OSRCI). This method uses GANs to artificially generate images that closely resemble the images in ID classes but belong to OOD. The classifier is then trained and tested with an additional class containing these images. A similar method, although with a few modifications, is followed in [60].

The instability in the training of GANs, the time taken for them to converge and the low possibility of reproducible results are the major limitations of using the GAN-based generative OOD detection methods. Moreover, most GAN-based methods are highly dependent on the reconstruction error for OOD detection, based on the assumption that an in-distribution image produces a lower reconstruction error than an OOD image. However, this idea does not work well to produce high-resolution images.

OpenGAN The OpenGAN method [61] deals with reducing the instability involved in the training GANs for generating high-resolution novel OOD images. While most of the previous GAN-based works concentrated on training the generator to produce the best OOD images, in this method, the focus is to get the best discriminator to learn to discriminate ID vs OOD images. The method uses a combination of outlier images and generated fake images to train a discriminator that can identify the novel (or 'unknown') OOD class. More importantly, the lightweight discriminator trained using the OpenGAN method can be retrofitted to the top of an existing multi-class classifier.

Tempered Mixup Data Augmentation One of the generative methods which do not use GANs is presented in [62]. The methods combine a novel form of the Mixup data augmentation [63] technique with an auxiliary loss function. The Mixup augmentation generates novel OOD images and the loss function makes the model to be less confident

towards images from this novel class. However, the method exceeds in performance only when compared to the methods that use an explicit novel class as an additional class for training.

2.4 Uses of OOD detection

Apart from making computer vision models safe, reliable and robust, a few more uses of out-of-distribution (OOD) detection are:

- **Saving energy on edge devices:** On an energy-constraint edge device, uploading every image taken to the cloud can be avoided to save energy. On such devices, using an OOD detector to identify and reject OOD images can help to prevent uploading images that are not worth classifying.
- **Prevent adversarial attacks:** Adversarial attack [11] refers to the intentional manipulation of input images intending to cause misclassification by the classifier. A few OOD detection methods such as [43] are proven to be useful for detecting adversarial input images.
- **Cleaner datasets:** OOD detection methods can be used to identify images that are corrupted or are outliers in a set of images. This can be applied while collecting images for making a dataset or on existing datasets to identify mislabelled or unusable images.
- **Active learning:** OOD detection can be used to select the most informative samples from a set of images for incremental learning of a classifier. Selecting such samples can improve the training efficiency of a classifier.
- **Trust in AI systems:** Having a reliable classifier that is more 'aware' of the open-world scenario and which makes reliable predictions can increase the trust in AI systems that use these classifiers.

Chapter 3

Out-of-Distribution Detection Methods

This chapter explains the out-of-distribution (OOD) detection methods used in this research on which various OOD detection experiments are performed. The list of the selected methods from the literature and the criteria used to select these are presented in Section 3.1. The selected methods from the literature and the proposed method developed in this research are explained in detail in 3.2.

3.1 Selecting Methods From Literature

A few of the OOD detection methods from existing literature are selected for experimentation in this research. The following criteria are used to shortlist the methods:

- **Retrofittable to existing classifiers** Retrofitting OOD detection methods to existing classifiers is useful in many practical scenarios where a classifier is already trained and deployed. This can also save time and computational resources as it eliminates the need to train a new model. This can be particularly beneficial if the model has to be trained on large datasets.
- **Do not require OOD images for calibration** An OOD detection method that requires OOD images to calibrate might show high accuracies when tested on similar OOD images. In an open-world setting, a classifier can encounter OOD images that are very different from the ones the model has been calibrated on. Furthermore, making a set of calibration images that can generalise any OOD can be impractical.
- **Different from each other** While shortlisting the methods, it also made sure that no two methods use similar concepts/assumptions to assign an OOD score for an input image. It also ensured that there is a spread between methods that use feature embeddings as well as the methods that primarily use output scores for OOD detection.

Table 3.1 shows the OOD detection methods shortlisted by following the above criteria, and a proposed OOD detection method called Entropy weighted nearest neighbour's distance (EnWeDi). The methods in the same category follow similar concepts of OOD detection and can be considered variations of one another. The results for the methods in bold are presented as the main set of results. The remaining methods are used in comparison experiments to gain deeper insights into OOD detection. All the methods and their variations are explained in detail in the next section (Section 3.2).

Type	Concept	OOD Method	Shortform used in this thesis	Relevant literature
Output score-based	Probabilities from softmax distributions	**Baseline using maximum softmax probabilty**	**Baseline MSP**	[1]
	Energy function on output scores	**Prediction probabilities' energy-based method**	**Energy**	[2]
	Input perturbations and temperature scaling	**Input perturbation-based ODIN**	**ODIN**	[3]
	Gradients from neural network	**Gradient space-based GradNorm**	**GradNorm**	[4]
Feature embeddings-based	Feature reconstruction error (FRE)	**Global (entire ID dataset) PCA based FRE method**	**Global PCA FRE**	[5]
		Per-class PCA based FRE method	Per-class PCA FRE	[5]
	Distance in feature space	**Distance to k nearest neighbours**	**DkNN**	[6]
		Distance to k nearest class centroids	Centroid-DkNN	concept from [29]
	Density in feature space	**Simplified Local Outlier Factor**	**SLOF**	concept from [7]
	Uncertainty and distance in feature space	**Entropy weighted nearest neighbour's distance**	**EnWeDi**	proposed method

TABLE 3.1: List of OOD detection methods used in this research. The results for the methods in bold are presented as the main set of results.

3.2 Out-of-Distribution Detection Methods

3.2.1 Maximum Softmax Probability : Baseline Method

A simple probability threshold-based method is proposed in [1] and has been used as a standard baseline for out-of-distribution (OOD) detection in several works ([6],[39],[4]) in the literature. The idea is to use the (softmax) output probabilities of a trained classification model to determine whether an input image is in-distribution (ID) or out-of-distribution (OOD). The method is based on the assumption that for a CNN trained on the ID dataset, the images from ID, in general, tend to have higher maximum probabilities. In contrast, for an out-of-distribution (OOD) image, the maximum of predicted probabilities is assumed to be lower, allowing for its detection. Figure 3.1 shows this idea. This method is called maximum softmax probability (MSP) and is used as the baseline method in this work.

The MSP is used as the OOD score (mentioned in Equation 2.1) and is given by:

$$MSP(x) = maximum(\{S_i(x) : i = 1, \ldots, C)\}) \ , \tag{3.1}$$

where $S_i(x)$ is the softmax probability of an input image x belonging to the i^{th} class (out of C classes). If the maximum softmax probability (MSP) is below a certain threshold, the image is considered OOD.

Being simple and computationally not expensive, the baseline method can be easily integrated with an existing classification model. However, it is also often observed that

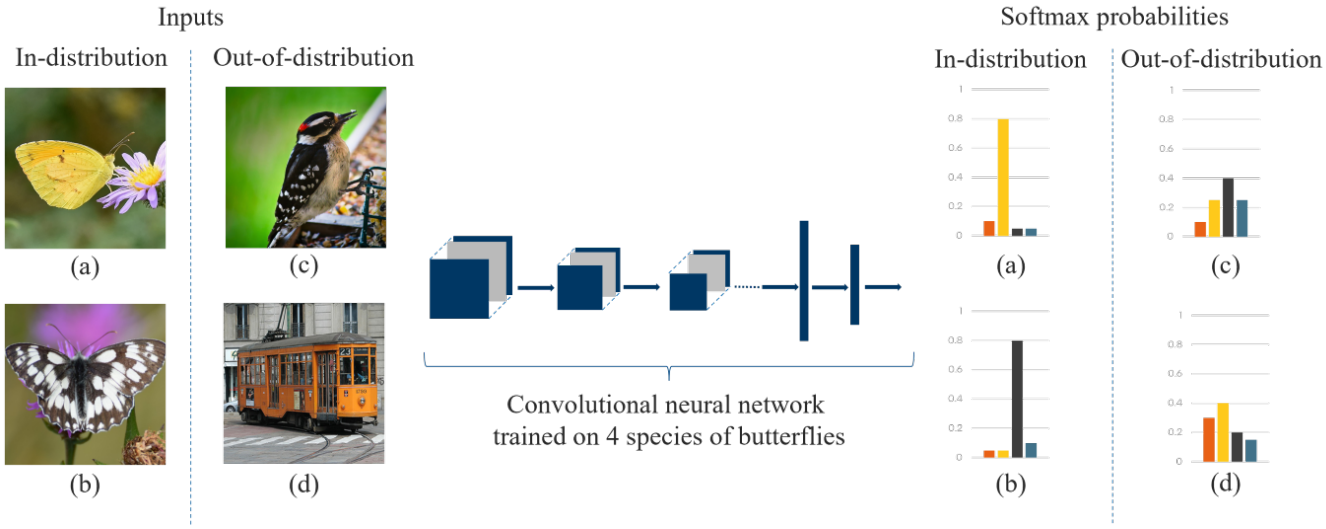


FIGURE 3.1: Assumption behind the maximum softmax probability-based baseline method

classifiers assign high softmax probabilities to OOD images and hence be woefully incorrect [10, 1, 11, 12]. Therefore, the underlying assumptions made are not always valid, making the baseline method less robust for OOD detection, thereby revealing room for better methods.

Variations in baseline methods Apart from the maximum softmax probability, the margin of the softmax probabilities i.e., the difference between the highest and the second highest softmax probability is also considered as OOD detection scores. Furthermore, the authors in [64] show that applying the concepts of these baselines on the logits instead of the softmax probabilities can, at times, give higher OOD detection accuracies. The results with these variations are compared in Section A.3 in the appendix.

3.2.2 Prediction Probabilities' Energy-based Method

As mentioned in the baseline methods (Section 3.2.1), softmax probabilities often suffer from overconfident posterior distributions for out-of-distribution (OOD) images [10, 1, 11, 12]. Hence, the OOD detection methods that rely on softmax probabilities also suffer from this phenomenon. The Energy-based OOD detection method from [2] circumvents this by using *energy scores*. The idea behind the Energy-based OOD detection method is to build an energy function that maps an input image x to a single value called energy. The energy value should be fairly similar among in-distribution (ID) images and OOD images while being significantly different between the two. Figure 3.2 shows this idea.

The energy-based OOD detection score is presented below. Consider a CNN classifier G trained on C classes, which maps an input image x to a set of C logit values using a function g(x). Then, the free energy $E(x;G)$ is given by:

$$E(x;G) = -T \cdot log \sum_{i=1}^{C} e^{g_i(x)/T} \ , \tag{3.2}$$

where $g_i(x)$ indicates the logit corresponding to the i^{th} class for the input image x. T is the Temperature parameter to perform temperature scaling [36] on logits. The concept of

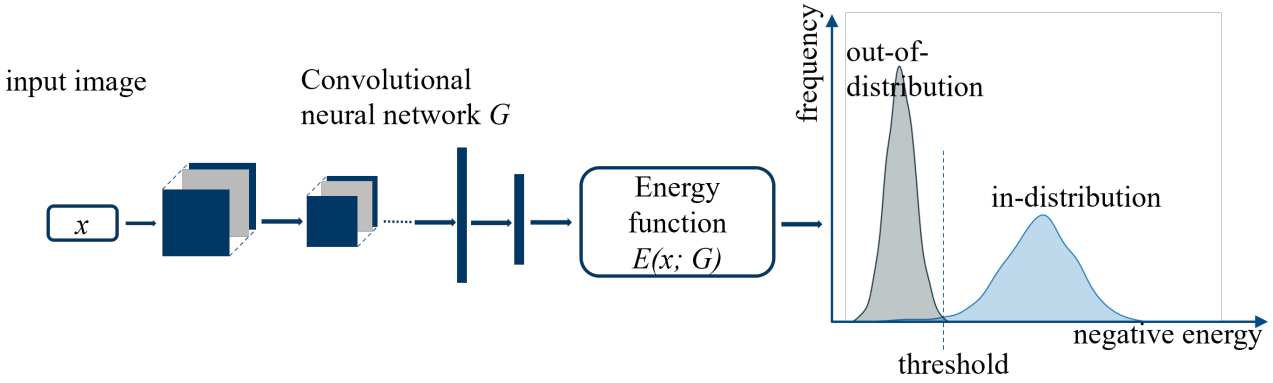


FIGURE 3.2: Concept of using an energy function for out-of-distribution detection

Temperature scaling is explained in Subsection 3.2.3.

3.2.3 Input Perturbation-based Method: ODIN

ODIN (Out-of-DIstribution detector for Neural networks) [3] is a method for out-of-distribution (OOD) detection that combines two concepts - temperature scaling [36] and adding small perturbations to the input. It was observed that adding small perturbations to the input images along with temperature scaling the output logits can make the ID and OOD images more distinguishable. The concepts of temperature scaling and adding perturbations as well as combining them to develop the ODIN method are explained below.

Temperature scaling Temperature scaling [36] technique is used to calibrate the output probabilities produced by a neural network by applying a so-called 'temperature' parameter T to the logits before the softmax function is applied to them. Consider a neural network G trained to classify C number of classes and gives output logits $g(x) = (g_1, ..., g_C)$ for an input image of x. The softmax probabilities with a temperature scaling factor of T are calculated as:

$$S_i(x;T) = \frac{e^{g_i(x)/T}}{\sum_{j=1}^{C} e^{g_j(x)/T}} \ . \tag{3.3}$$

When $T > 1$, the softmax probabilities become smoother and less peaky, resulting in a less confident and better-calibrated classifier. As mentioned in [36], performing temperature scaling makes the softmax probabilities to be more reliable (better calibrated).

Input perturbations In addition to temperature scaling, small perturbations are added to an input image to increase the maximum softmax score. These small perturbations tend to have a stronger effect on the in-distribution images than on out-of-distribution images, making it easier to detect the latter. The perturbations are computed by back-propagating the gradient of the cross-entropy loss with respect to the input image. The technique is adapted from [65] where perturbations are added to the inputs to create adversarial images. Adding perturbations to an input image x is given by:

$$\widetilde{x} = x - \varepsilon \cdot sign(-\nabla_x \ log(S_{\hat{i}}(x;T)) \ , \tag{3.4}$$

where $S_{\hat{i}}$ is the maximum softmax probability for x and ε is the perturbation magnitude.

Combining input perturbations with temperature scaling The following steps explain the ODIN method :

- **Step 1: Infer input image** Infer the input image on the given classifier G without any perturbations to get the output logit scores. The classifier, (in this case, a CNN model) was trained on an in-distribution dataset.
- **Step 2: Temperature scaling** On the output logits, apply temperature scaling using Equation3.3 .
- **Step 3: Generate an image with perturbations** Apply softmax on logits from step 1 and calculate the maximum softmax probability and calculate the cross entropy loss on the logits obtained from step 2. Apply Equation3.4 to get $\widetilde{x}$, the input image with perturbations added.
- **Step 4: Prediction probabilities for $\widetilde{x}$** Apply steps 1 and 2 on the image with perturbations ($\widetilde{x}$) and apply softmax on the output scores obtained. The maximum of this calibrated softmax probabilities is used as the OOD score for the input image x.

The OOD score so obtained is compared with a threshold to determine if the input image x is in-distribution or out-of-distribution. The general approach to threshold calculation is explained in Subsection 2.1. The parameters T and ε are optimised while determining this decision threshold.

3.2.4 Gradient Space-based Method: GradNorm

GradNorm [4] utilises the gradients extracted from the output layer of a neural network to calculate the out-of-distribution (OOD) score. Specifically, gradients are backpropagated from the Kullback-Leibler divergence (KL divergence) [41] between the softmax output and uniform distribution. The idea behind GradNorm is that for an in-distribution image the prediction scores tend to be high for only one of the classes. This makes the prediction scores far from being a uniform distribution, making the aforementioned Kullback-Leibler divergence large. The gradient norm of the Kullback-Leibler divergence would be higher for an ID image than that for an OOD image.

GradNorm score calculation For a given input image x, consider a classifier G which outputs predicted scores $q = \{q_i\}$. Then the Kullback-Leibler divergence (KL divergence) [41] can be used to quantify how close the predicted scores $\{q_i\}$ are to a reference set of scores given by $p = \{p_i\}$. This is given by

$$D_{KL}(p||q) = \sum_i p_i log(\frac{p_i}{q_i}) = -\sum_i p_i log(q_i) + \sum_i p_i log(p_i) = H(p,q) - H(p) \ . \tag{3.5}$$

In the above equation, H(p,q) is the cross entropy between the distributions p and q.

The reference set of scores p is set to be a uniform distribution given by $p = u = [\frac{1}{C}, \frac{1}{C}, ..., \frac{1}{C}$, where C is the number of classes the classifier has been trained on. The output prediction scores (logits) of the classifier are represented as $g(x) = (g_1, ..., g_C)$ for an input image of x. Furthermore, the softmax function is applied to these prediction scores, along with temperature scaling [36]. Now Equation 3.5 becomes :

$$D_{KL}(u||softmax(g(x))) = -\frac{1}{C}\sum_{c=1}^{C} log\frac{e^{g_c(x)/T}}{\sum_{i=1}^{C} e^{g_i(x)/T}} - H(u) \tag{3.6}$$

where the first term is the cross-entropy loss between softmax prediction probabilities and the uniform distribution u.

For the GradNorm score, D_{KL} is differentiated with respect to the set of network parameters at the final layer. If the set of network parameters is w, then the OOD score is given by :

$$s(x) = \| \frac{\partial D_{KL}(u||softmax(g(x)))}{\partial w} \|_p \ , \tag{3.7}$$

where $\| \ . \ \|_p$ denotes the L_p norm.

GradNorm can be implemented by calculating the cross-entropy loss between the predicted softmax probabilities and a uniform vector as the target probabilities.

3.2.5 Feature Reconstruction Error-based Methods

Feature reconstruction error (FRE) methods for out-of-distribution (OOD) detection work by reconstructing the features of an image using a fitted transformation and then calculating the reconstruction error. The reconstruction error is used as the OOD score in Equation 2.1 to determine if an input image belongs to ID or OOD. The assumption is that samples from the in-distribution will have a low reconstruction error, while samples from OOD distributions will have a high reconstruction error.

Principal component analysis- A brief introduction Principal component analysis (PCA) [66] is a dimensionality reduction technique that is used to transform high-dimensional data into a lower-dimensional representation while retaining a set fraction of original information. PCA works by identifying the directions in the data that have the highest variance (i.e., the directions that capture the most information about the structure of the data). These directions are called the 'principal components', and the transformed data is a linear combination of these components.

The process of PCA can be summarised as follows:

- **Standardising of data** The data is centred by subtracting the mean from each feature. This ensures that the first principal component is aligned with the direction of maximum variance, rather than the direction of the mean.
- **Covariance matrix calculation** The covariance matrix is calculated to capture the relationships between the features, i.e., how the features of the data set are varying from the mean with respect to each other.
- **Eigenvalue decomposition** The covariance matrix is decomposed into its eigenvalues and eigenvectors. The eigenvectors represent the directions of maximum variance, and the eigenvalues represent the magnitudes, i.e., the importance of the respective variances.
- **Dimensionality reduction** The number of dimensions is reduced by selecting the top k eigenvectors, where k is the desired dimensionality of the transformed data.

3.2.5.1 Global Principal Component Analysis Feature Reconstruction Error Method

The global PCA-based feature reconstruction error method is based on modelling the subspace of the intermediate features produced by a DNN. In this approach, the PCA model trained on in-distribution images is used to reduce the dimensionality of an input image's

feature embedding and produce a feature embedding with fewer dimensions. The same PCA model is used to reconstruct the original feature from its low-dimension representation. Feature reconstruction error (FRE) is the Euclidean distance between the original feature and its reconstructed version. If the reconstruction error exceeds a certain threshold, it is assumed that the input data is an out-of-distribution sample.
This idea is shown in Figure 3.3.

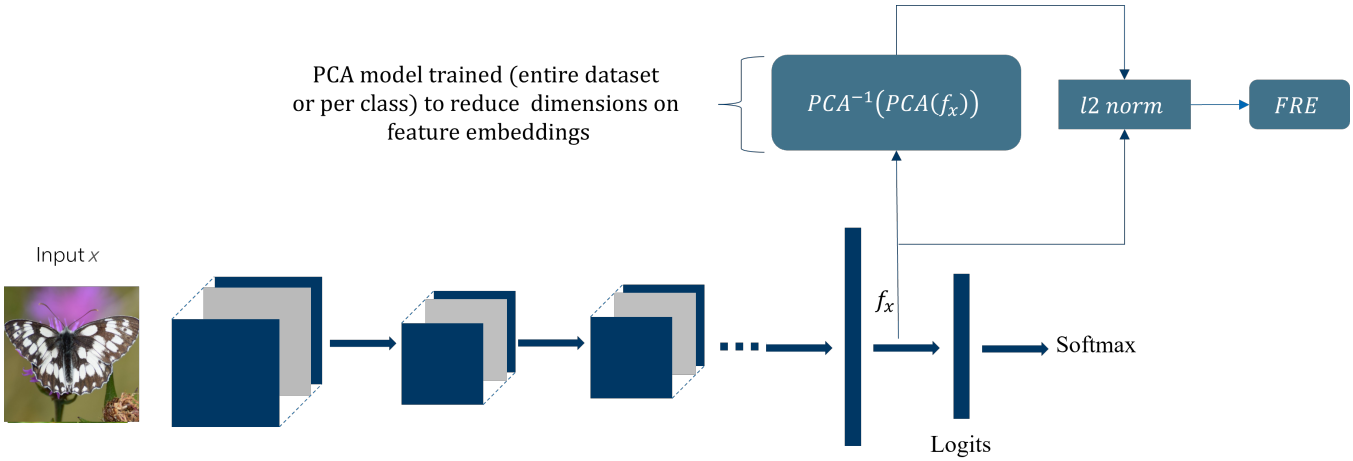


FIGURE 3.3: Concept of using feature reconstruction error using PCA for OOD detection. PCA and PCA^{-1} in the figure are the forwards and inverse transformations for dimensionality reduction respectively

The following steps explain the Global PCA-based FRE.

- **Step 1: Feature Embeddings Extraction** Feature embeddings of the images from the training split of the ID dataset are extracted from the penultimate layer of the deep neural network. The deep neural network already trained to perform the image classification task on the ID dataset can be used. However, for tasks like anomaly detection, a DNN pre-trained on ImageNet can be used as well.

- **Step 2: PCA model Training** A PCA model is trained to perform linear dimensionality reduction on these feature embeddings while retaining the set level of variance. The fraction of the variance of the original data that should be retained in the reduced feature embeddings is a hyperparameter. During this training, the PCA model learns the transformation that maps a feature embedding into a lower-dimensional one. In other words, it learns to model the subspace of the feature embedding. During this process, the PCA model, learns the inverse transformation as well, i.e., to 'reconstruct' a lower dimensional feature embedding into its original dimensionality.

- **Step 3: Test phase** During the testing phase, the feature embedding f_x of the input image x is extracted as mentioned in step 1. To this feature embedding, the dimensionality reduction transforms τ of the trained PCA model is applied to get the reduced feature embedding f_red_x. f_red_x has fewer dimensions than f_x. The inverse transformation τ^{-1} is applied on this reduced feature embedding to produce the reconstructed feature embedding f_rec_x. f_rec_x is of the same dimensions as the original feature embedding f_x. The FRE can then be calculated as the $l2$ norm of the difference between the original f_x and reconstructed feature embedding f_rec_x.

FRE is given by the equation:

$$FRE(x) = \parallel f_x - \tau^{-1}(\tau(f_x)) \parallel_2 . \tag{3.8}$$

The FRE is directly used as the OOD score for image x. The PCA model has learned the forward and inverse transformations using the images from the ID dataset. Hence, the assumption is that the same transformations cannot reconstruct the feature embedding of an OOD image, leading to a higher FRE.

In this method, during the training of the PCA model, all the images (irrespective of the classes they belong to) are considered as a single entity. Hence global modelling is done by the PCA model, where the forward and inverse transformations are learned for the images in the training set. This approach can be useful when the number of images per class is relatively small to the number of dimensions of the feature embeddings. It can also be used when there the class labels are not available.

3.2.5.2 Per-Class Principal Component Analysis Feature Reconstruction Error Method

While global modelling of the entire training set into a single PCA model has its merits, it may not adequately model the sub-spaces of the individual classes present. In this method of FRE, each PCA model is trained on the feature embeddings of a single class.

The method follows similar steps as the Global PCA FRE method with the following changes:

- **Step 1:Extracting the feature embeddings** Same as done for the Global PCA FRE method.
- **Step 2: Training the PCA models** The feature embeddings are grouped based on their respective classes. A PCA model per each class is trained on feature embeddings belonging to the respective class. The required variance to be maintained (or the number of dimensions to have) in the reduced feature is kept constant across all the models.
- **Step 3: Test phase** During the test phase, the feature embedding f_x of an input image x undergoes the reconstruction process through all the PCA models. The feature reconstruction error (FRE) is calculated at all the PCA models. The lowest of all the FREs is taken as the final OOD score.

3.2.6 Feature Embeddings' Distance-based Methods

The core concept in all the distance-based methods is that the OOD samples are relatively far away from a known set of ID samples in a feature embedding space. Samples are often represented by the feature embeddings extracted from a model.

3.2.6.1 Short introduction to the k-nearest neighbours algorithm

K-nearest neighbour (kNN) is a machine learning algorithm for classification as well as for regression and is often used due to its simplicity. The basic idea behind kNN in a classification task is to assign a label to an unseen sample based on what its closest neighbours (from the training/known set) are.

Here are the major steps involved in performing a kNN search:

- Data preparation: The algorithm starts with a training data set, which contains labelled data points. These labelled data points are represented in a space where each dimension corresponds to a feature of the points.
- Distance calculation: To determine the closest neighbours of an unseen data point, the algorithm calculates the distance between that data point and each of the data points in the training set. The distance is usually calculated using a distance metric such as Euclidean distance, Manhattan distance, or any other form of Minkowski distance.
- k-nearest neighbours selection: The algorithm selects the 'k' number of data points from the training set that is closest to the unseen data point, based on the calculated distances. 'k' here is the number of neighbours to consider when making a prediction and is a hyperparameter.

3.2.6.2 Deep k-Nearest Neighbours Method

The 'deep k-Nearest Neighbours for OOD detection' (DkNN) method presented in [6] is based on the kNN algorithm. In this method, the k^{th} nearest neighbour distance between the feature embeddings of an input image and the images from the ID training set is computed. This k^{th} distance is compared with a threshold to determine if an input image is OOD or not. Figure 3.4 shows the concept of this method.
The DkNN method uses the Facebook AI Similarity Search (Faiss) [44] - a library for efficient similarity search for performing the kNN search, making it plausible to be used for OOD detection even on large datasets.

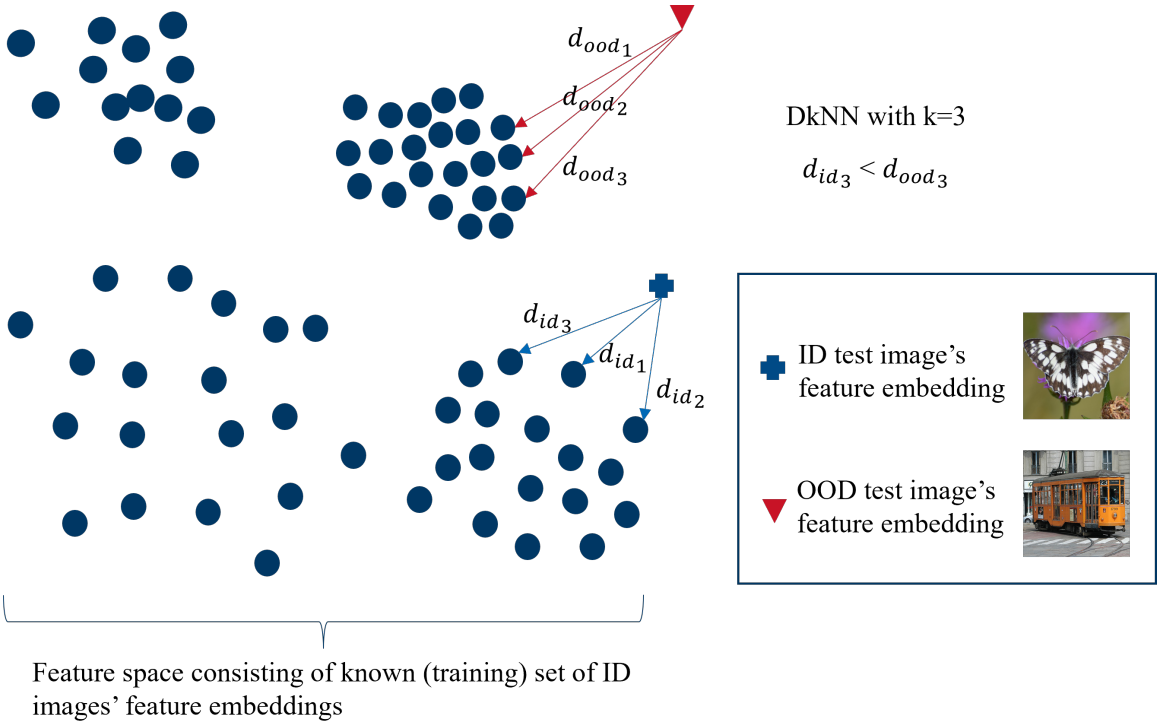


FIGURE 3.4: Deep k-nearest neighbours method for OOD detection with an arbitrary k of 3.

The DkNN method is explained below:

- **Step 1: Feature extraction** For each image in the training split of the in-distribution dataset, the feature embeddings are extracted from the penultimate layer of a model. The model was trained on the training set of the ID dataset.

- **Step 2: Normalise feature embeddings** The feature embeddings are normalised by dividing each feature embedding vector by its respective 2nd-order vector norm (vector Euclidean norm). Although not mandatory, it has been shown by the authors of DkNN [6] that normalising feature embeddings lead to a significant improvement in OOD detection.

- **Step 3: Indexing the training feature embeddings** The feature embeddings of the training set images are indexed and added to the kNN search space using the Faiss library. The dimensions of the search space are equal to the length of the feature embeddings. Note that feature embeddings are not tagged with their respective class labels. All the training set feature embeddings are treated as belonging to a single set - in-distribution feature embeddings.

- **Step 4: Setting the decision threshold** As mentioned in Subsection 2.1, before the test phase, a distance threshold is calculated. The threshold is set in such a way that a large number (for example 95%) of a set of known in-distribution images' distance to the k-closest image in the training set is below the threshold. The set of validation images from the ID dataset can be used for this purpose.

- **Step 5: Test phase** During the test phase, the feature embedding of the test image is extracted and normalised. A kNN search on the test image's feature embedding is performed which returns the distances to the k-nearest neighbours. These distances are then sorted in ascending order and the last (k^{th}) distance is picked as the OOD score for the test image. The kNN search is performed using the Faiss library and can be done on a GPU for faster execution.

- **Step 6: Decision** The k^{th} distance is compared with the above-mentioned threshold. If the OOD score of the test image is greater than the threshold, it is classified as an OOD image.

This deep k-nearest neighbours method from the literature without any modifications will be referred to as **DkNN** in short in this thesis document.

Variations of the deep k-nearest neighbours method Apart from using the distance to the kNNs, the following metrics were also tested for suitability as OOD score in this work :

- The mean of distances to the kNNs.

- Average of the distance to the nearest and farthest neighbour out of k neighbours.

- Other statistical representations such as mode, the median and standard deviation on the distances to the k- nearest neighbours.

The DkNN method with these metrics as OOD score is compared in Table A.5 in the appendix.

3.2.6.3 Centroid-DkNN

The Centroid-DkNN is based on the concepts used in the nearest class mean (NCM) classifier [29]. In this method, a per-class average of the normalised feature embeddings is used instead of feature embeddings of all the images in the training set. Each averaged feature embedding represents the centroid of the respective class. The other steps (steps 3-6) remain the same as the DkNN method from the literature. The Centroid-DkNN method is also called the k-nearest prototypical neighbours' distance method.

This drastically reduces the number of feature embeddings in the kNN space. This means that performing a kNN search is faster and requires less memory on the hardware, making it appealing for OOD detection on low-power embedded systems. The concept of using centroids (means) of feature embeddings from each class for OOD detection is shown in Figure 3.5. Note that the labels of training data are required to calculate these centroids per class. This can be done after the training of the classifier with the same training data. However, once the centroids are found, during the test phase, the labels are no longer required.

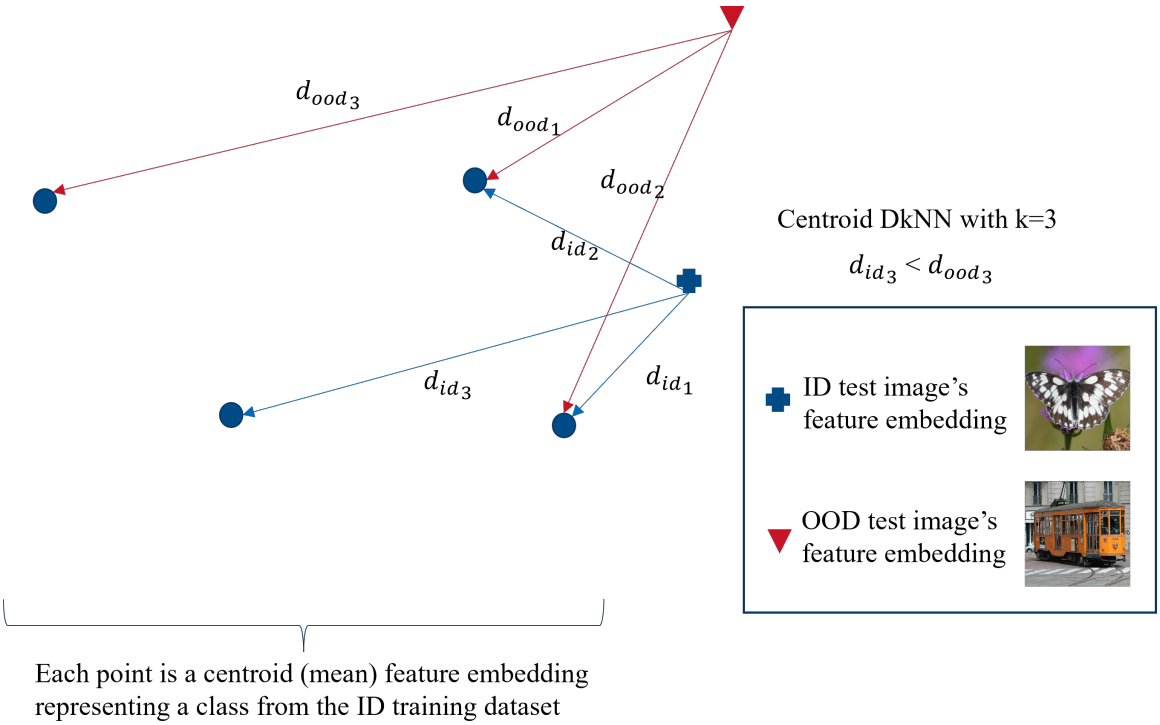


FIGURE 3.5: Using centroid of each class's feature embedding for OOD detection.

3.2.7 Density-based Outlier Factor Methods

Methods that use only distance to the nearest neighbours for OOD detection might not perform as expected when the different classes have varying levels of density clusters in the feature space. For instance, in Figure 3.6, the feature embeddings of images from various classes' have varying levels of density in feature space. Both the points TI and TO would have similar distances to their nearest neighbours. The feature embeddings in C_2, are relatively at a smaller distance from their respective nearest neighbours. However, TO does not show this property with feature embeddings from C_2. On the contrary, TI and the feature embeddings in class C_4 are located at similar distances to their respective

nearest neighbours. In other words, TI and the feature embeddings in C_4 have similar densities in feature space. Hence it is more probable that TI might be an ID data point. In this case, the point TO should have a higher score for OOD than that for TI. This can be achieved using an OOD detection method that combines the density of the k-nearest neighbours (kNNs) along with the distance to those kNNs.

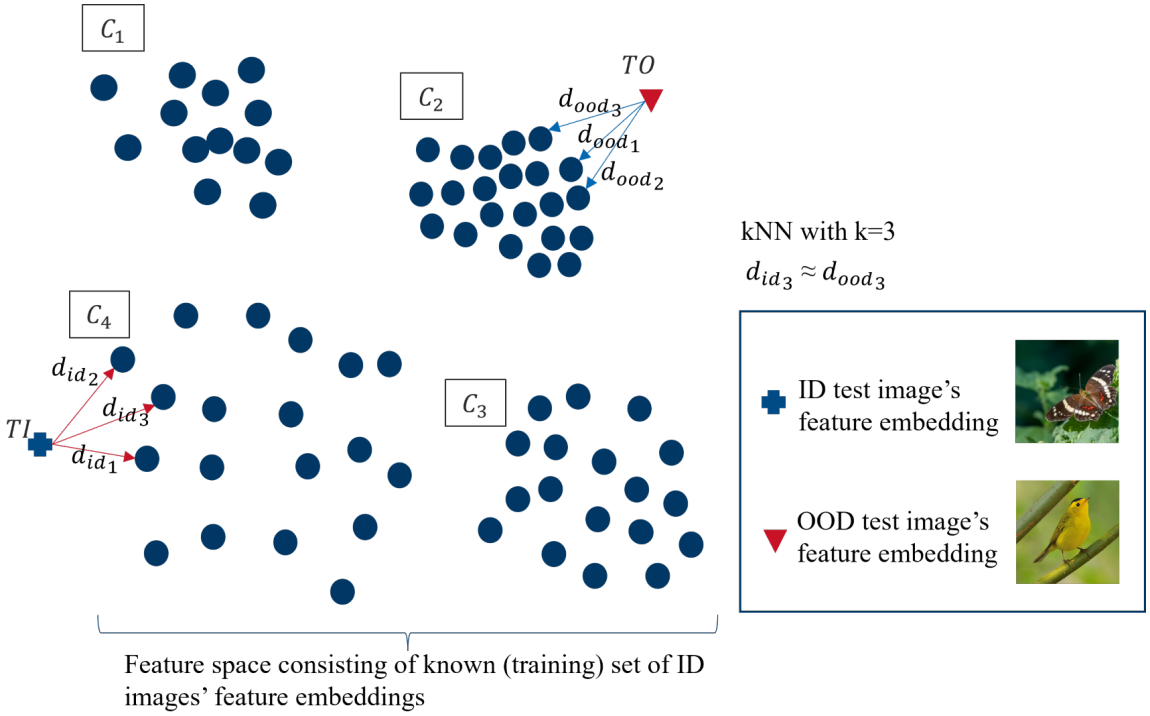


FIGURE 3.6: An example of where using only the distance to the nearest neighbours may not be useful in distinguishing between OOD image and ID image

3.2.7.1 Simplified Local Outlier Factor

A simplified version of the LOF algorithm [31] is mentioned in [7].

For a test image x, let f_x be the normalised feature embedding. If the set of k nearest neighbour feature embeddings (from the training set) of f_x is given by N_{x_k}, then the simplified LOF (SLOF) of f_x is calculated as

$$SLOF_k(f_x) = \frac{\sum_{f_i \in N_{x_k}} \frac{kdist(f_x,k)}{kdist(f_i,k)}}{k} , \tag{3.9}$$

where $kdist(f_i, k)$ is the distance between f_i and its k^{th} nearest neighbour. In this work, it is the Euclidean distance between f_i and its k^{th} nearest neighbour.

The ratio $kdist(f_x, k)/kdist(f_i, k)$ is a measure of the relative k^{th} distance between f_x and f_i and hence the simplified LOF is, in a way, measuring the average relative distance between the feature embedding f_x and its k neighbours.

Two other density methods - Local outlier factor (LOF) and Local distance-based outlier factor (LDOF) were also evaluated in this work. However, to prioritise the most relevant findings, these methods are moved to Section A.4 of the appendix.

3.2.8 Entropy Weighted Nearest Neighbour Distance Method

For an OOD image that is visually similar to ID images, it was observed that the distance between its feature embedding and the k^{th} nearest neighbour (of feature embeddings from the ID training set) is at times similar and a few times even smaller than that of an ID image. In such cases, the underlying assumption of the DkNN method holds weak or does not hold. Additionally, in such cases, finding the distance threshold above which an image is classified as OOD becomes difficult.

To overcome this limitation of the DkNN method, a kNN-based non-distance metric is explored initially in this research. In this regard the following hypothesis is made: *For an OOD image, even though the distance to its k-nearest neighbours may be comparable to that of an ID image, the ID class to which k-nearest neighbours belong is diverse and spread out. While, for an ID image, the majority of the k-nearest neighbours belong usually to a single class (or relatively fewer classes).* This is illustrated in Figure 3.7. In this figure, the distance to k^{th} nearest neighbour is similar for ID and OOD feature embedding. For the ID feature embedding, most of the nearest neighbours (4 out of 5) belong to a single class of ID, while for the OOD feature embedding the nearest neighbours are (almost uniformly) spread across 3 different classes of ID.

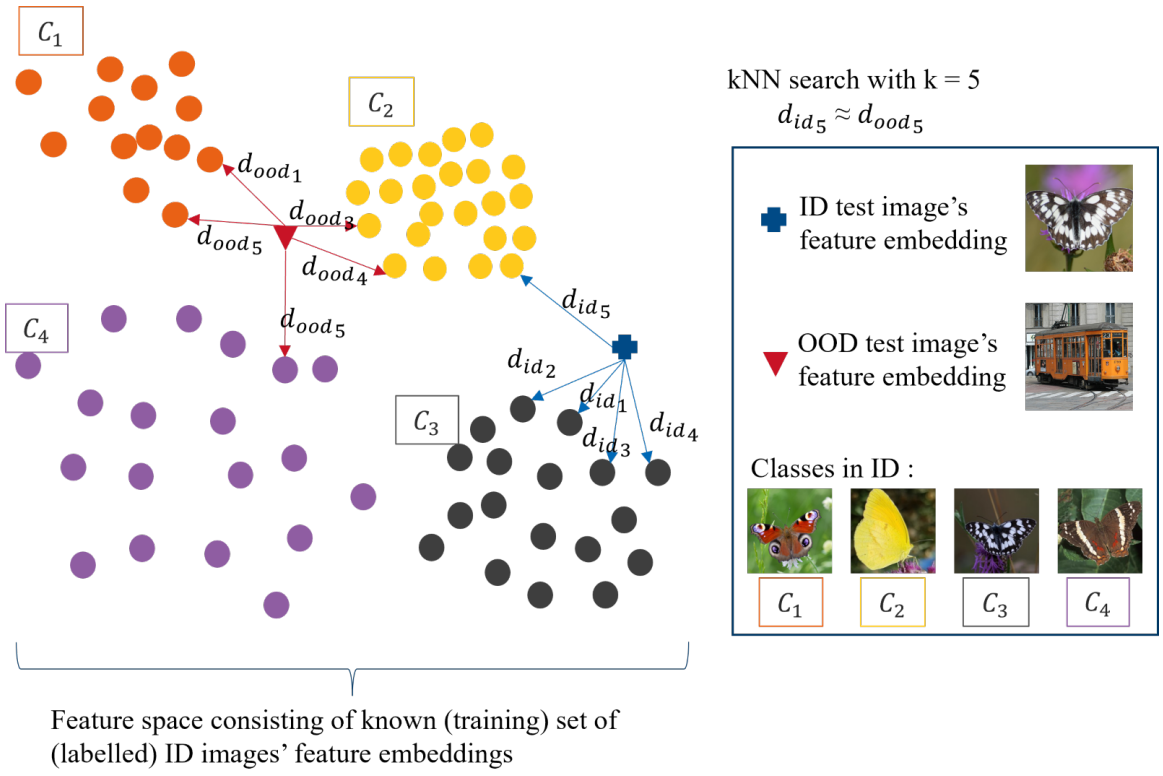


FIGURE 3.7: Hypothesis behind using entropy among k-nearest neighbours in feature space for OOD detection. ID and OOD image's feature embedding being at similar distances to their respective k^{th} nearest neighbour in feature space. In such cases, using the kNN distance as the OOD score is not useful. Notice that the nearest neighbours for OOD feature embedding are spread across 3 different classes, while for ID most of them belong to a single class.

Initial idea: The k nearest neighbour entropy OOD detection method From the above-mentioned observation, the diversity in the nearest neighbours of a test image's feature embedding, if quantified, can be used as an OOD score. Based on this idea, the k nearest neighbour entropy (KENT) OOD detection method is proposed. In this method, following a kNN search for a test image's feature embedding, a measure of uncertainty on the classes to which the k-nearest neighbours belong is calculated using Shannon entropy [67]. In general, entropy gives a measure of the amount of uncertainty or randomness in a probability distribution. For OOD detection, the k nearest neighbour entropy (KENT) between a test image's feature embedding f_x and its k nearest neighbours in feature space is calculated by:

$$KENT_k(f_x) = -\sum_{c \in C_{T_k}} \frac{n_c}{k} log_k(\frac{n_c}{k}) \ , \tag{3.10}$$

where C_{T_k} denotes the set of unique classes the k-nearest neighbours of f_x belong to and n_c is the number of neighbours (from the set of k-nearest neighbours) belonging to the class c (from the set of C_{T_k} classes).

The assumption is that for an ID image x, the $KENT(f_x)$ would be low indicating that it likely belongs to a particular class in the ID, and vice versa for an OOD image.

An advantage of the KENT method is that the OOD scores are bounded in the range of [0,1]. A bounded OOD score is also easier to interpret, as the maximum and minimum OOD scores are known. Furthermore, a heuristic requirement based on what makes an input ID or OOD can be easily translated into an OOD score threshold without performing any experiment. For instance, consider a simple heuristic such as *"classify an input image as ID if it has no more than two unique classes in the set of 9 nearest neighbours in the feature space"*. With the KENT method, this can be mapped into OOD scores of [0, 0.31] for ID and (0.31, 1] for OOD.

Entropy weighted nearest neighbour distance method The entropy calculated from Equation 3.10 can be already used as an OOD score. However, a more robust OOD detection method that leverages combining the distance information with the entropy of the nearest neighbours of a test image's feature embedding is devised. This proposed method is called the Entropy weighted nearest neighbour's distance (EnWeDi).

In a set of k-nearest neighbours of a f_x, the EnWeDi considers distance for the first k_d nearest neighbours and the entropy of the remaining neighbours. However, this leads to having two hyperparameters - k_d and k. An assumption that the distance to the nearest neighbour is already a robust estimation of the OODness of x can be made. Hence the hyperparameter k_d can be set to 1 thereby reducing the number of hyperparameters to 1. With this assumption, the EnWeDi for a feature embedding f_x of an input image x is given by:

$$EnWeDi_k(f_x) = kdist(f_x, 1) * (1 + kent^*_{(2,k)}(f_x)) \ , \tag{3.11}$$

where $kdist(f_x, 1)$ is the distance between f_x and its nearest neighbour in feature space. $kent^*$ is a modification of the $kent$ from Equation 3.10 and is given by

$$KENT^*_{k_e,k}(f_x) = -\sum_{c \in C_{T_{k_e,k}}} \frac{n_c}{k - k_e + 1} log_k(\frac{n_c}{k - k_e + 1}) \ , \tag{3.12}$$

where k_e indicates from which number of nearest neighbours onwards KENT must be calculated. As mentioned above, it is set to 2 for the EnWeDi method. $C_{T_{k_e,k}}$ is the set of unique classes that the k-nearest neighbours of f_x excluding the first $k_e - 1$ neighbours belong to. n_c is the number of neighbours belonging to the class c (from the set of $C_{T_{k_e,k}}$

classes).

Intuition behind the EnWeDi method In practice, the EnWeDi performs a kNN search in the feature space just like the DkNN method. After the kNN search, the distance to the nearest neighbour is taken and the entropy ($kent^*$) of the remaining neighbours excluding the nearest neighbours is calculated. Intuitively, for a feature embedding f_x, the EnWeDi method increases the distance to the nearest neighbour by scaling it up by a factor in the range of [1,2] ($kent^*$ ranges from 0 to 1). Similar to the $kent$ value, for an OOD image, the value of $kent^*$ is assumed to be usually higher. Hence the distance to the nearest neighbour, which is also assumed to be usually higher for OOD, is scaled up with a factor >1. However, for an ID image, the $kent^*$ value is assumed to be smaller than that of an OOD image. In such cases, the distance (which is assumed to be smaller for an ID input) is 'weighted' with a relatively smaller factor. It is to be noted that the $kent^*$ can be as low as 0 as well, in which case, the OOD score is the same as the distance to the nearest neighbour of f_x. In this way, the EnWeDi aims to overcome the limitation of the DkNN method whose OOD scores are solely based on distance. The hypothesis behind the EnWeDi method is validated in Section 6.4.

It is to be noted that the advantage of using the Entropy weighted nearest neighbour distance as an OOD score can be seen only if $k > 2$. When $k = 1$, the $kent^*$ part in Equation 3.11 is set to 0 as the set of points (feature embeddings) on which entropy is to be calculated is empty. When $k = 2$, $kent^*$ becomes 0 irrespective of the input. Hence, for $k \leq 2$, the EnWeDi method is the same as the DkNN method with $k = 1$.

Chapter 4

Methodology

Section 4.1 explains the proposed approach in which the difficulty of an out-of-distribution detection problem is quantified in this work. The construction of in-distribution (ID) and out-of-distribution (OOD) datasets to test the methods under various aspects of OOD detection is described in Section 4.2. Section 4.3 describes the experiment setup and Section 4.5 lists the set of experiments performed to answer the research questions. The metrics on which the OOD detection methods are evaluated are presented in Section 4.6.

4.1 Domain Similarity Score as a Measure of OOD Detection Difficulty

In this work, the out-of-distribution (OOD) detection methods are evaluated on various OOD datasets each having a different level of OODness from the in-distribution (ID) dataset. An OOD dataset, depending on the degree of domain dissimilarity with the ID dataset, can be categorised into 'near-OOD', 'intermediate-OOD' or 'far-OOD'. An image from the 'near-OOD' dataset is highly similar (visually) to the ID images, making it difficult for the OOD detection methods to detect them.

In this work, the concept of domain similarity from [68] is used to numerically quantify how visually dissimilar an OOD dataset is, concerning the ID dataset. A thorough search of the relevant literature yielded little to no OOD detection research which used domain similarity from [68] as a measure of OOD detection difficulty.

Domain similarity of two datasets S, T from different distributions can be treated as the least amount of total work needed in moving all the images of dataset S to match the distribution of dataset T. This definition of domain similarity between an in-distribution dataset S and an out-of-distribution dataset T is measured by the earth mover's distance (EMD) [69] between the feature embeddings of the images from the two datasets. It is calculated as follows:

- **Distance between two classes** The distance between two classes m (from dataset S) and n (from dataset T) is calculated as the Euclidean distance between their centroids (mean of feature embeddings). If the set of images in m and n classes is I_m and I_n, and the number of images in m and n classes is represented by $|I_m|$ and $|I_n|$ respectively, then the Euclidean distance between the centroids of classes m and n is given by:

$$d(m,n) = \| \frac{\sum_{i=1}^{|I_m|} f_{I_{m_i}}}{|I_m|} - \frac{\sum_{i=1}^{|I_n|} f_{I_{n_i}}}{|I_n|} \|_2 , \tag{4.1}$$

where $f_{I_{m_i}}$ and $f_{I_{n_i}}$ are the feature embeddings of the i^{th} image in the classes m and n respectively.

- **Distance between two datasets** If there are M and N number of classes in datasets S and T respectively, then the distance $D(S,T)$ between them is given by :

$$D(S,T) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} F_{i,j}\ d(m_i,n_j)}{\sum_{i=1}^{M}\sum_{j=1}^{N} F_{i,j}}\ , \tag{4.2}$$

where $d(m_i,n_j)$ is the Euclidean distance between the i^{th} class m_i of the dataset S and j^{th} class n_j of the dataset T and is given by Equation 4.1. The optimal flow $F_{i,j}$ corresponds to the least amount of total work by solving the EMD optimisation problem [69].

- **Domain similarity** The domain similarity ψ between two datasets S and T with a distance $D(S,T)$ between them is given by:

$$\psi(S,T) = e^{-\gamma D(S,T)}\ , \tag{4.3}$$

where γ is a scaling factor for the distance and is a hyperparameter. It is set to 0.01 in this work.

The feature embeddings used for domain similarity need to be capable of representing the images' high-level information in an all-encompassing unbiased manner as much as possible. The CNN from which the feature embeddings are extracted must have been trained on a wide varied range of image classes. Hence, for the calculation of domain similarity score, the feature embeddings are extracted from the penultimate later (with global average pooling) of an EfficientNetV2-M [70] trained on the large dataset ImageNet-21K [8].

4.2 Preparing the In-Distribution and Out-of-Distribution Biodiversity Datasets

4.2.1 Imbalance Factor of a Dataset

In many real-world scenarios, the occurrence of certain objects is less frequent than others. Hence class imbalance is an inherent property in the real world (and not just of biodiversity). Hence before preparing the in-distribution (ID) and out-of-distribution (OOD) datasets, it is important to quantify them to build the right set of OOD experiments on them and to understand any patterns in their OOD detection results.

The ID dataset is measured in terms of dataset size (number of images) and class imbalance. In this work, the imbalance factor is used to quantify the disproportion between the number of images per class in an in-distribution dataset. The calculation of the class imbalance factor is explained below.

The imbalance factor is adapted from the Shannon diversity index-based balance factor used in [71] and is given by:

$$imbalance\_factor\ (D) = (1 - \frac{-\sum_{i=1}^{K}(\frac{|c_i|}{n} log(\frac{|c_i|}{n}))}{logK})\ , \tag{4.4}$$

where n is the total number of images in the dataset D, K is the number of classes and $|C_i|$ indicates the number of images in the i^{th} class. The range of the factor is [0, 1), and a perfectly balanced dataset such as ImageNet [8] will have an imbalance factor of 0.

4.2.2 Overview of the Datasets

The OOD detection methods are tested on ID datasets made from three biodiversity datasets - Naturalis-Papilionidae [72], CUB-200-2011 [73] and the iNaturalist 2017 [74]. Each dataset covers different levels of fine-grainedness, class imbalance, and dataset size.

A short introduction about the three publicly available biodiversity datasets from which ID and a set of OOD datasets are derived is given below.

Naturalis-Papilionidae Dataset Naturalis-Papilionidae [72] is a collection of specimens of butterflies belonging to the Papilionidae family and is provided by the Naturalis Biodiversity Center. It contains 8243 images that can be grouped into 3 hierarchical levels – genus, species, and subspecies. In this work, the images are grouped into 112 fine-grained classes based on the subspecies they belong to. The dataset is highly imbalanced as shown in Figure 4.1 and has an imbalance factor of 0.3057 (using Equation 4.4).

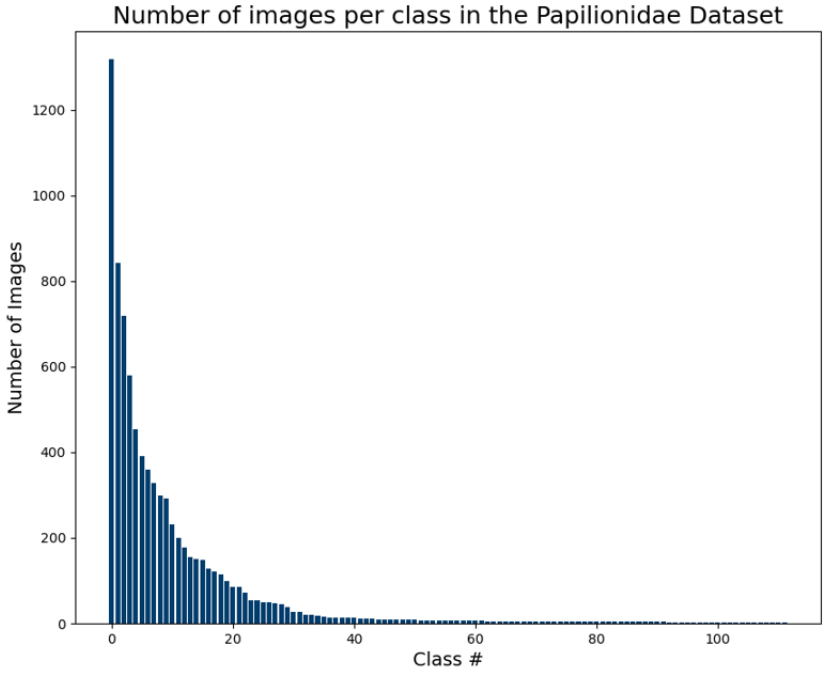


FIGURE 4.1: The distribution of the number of images per class in the Papilionidae dataset

CUB-200-2011 Dataset The Caltech-UCSD Birds-200-2011 (CUB-200-2011) dataset [73], also known as CUB-200, contains 11,788 images of 200 bird species. With similar-looking bird classes, the dataset is also fine-grained. However, the CUB-200 is a (nearly) balanced dataset (imbalance factor < 0.001) with approximately 60 images per class. The fewer number of images per class makes it challenging to model the training data distribution and generalise well enough.

iNaturalist Dataset The iNaturalist species classification and detection dataset [74] is one of the largest biodiversity datasets publicly available with 675,170 images. The

images are grouped into 13 super categories such as Plantae (plants), Insecta (insects), Aves (birds), Mammalia (mammals), and so on. The images are also grouped into 5,089 fine-grained classes (each representing a species), with a significant number of visually similar classes. In this work, the iNaturalist dataset from the year 2017 is used.

A few examples from the fine-grained classes of the three datasets are shown in Figure 4.2.

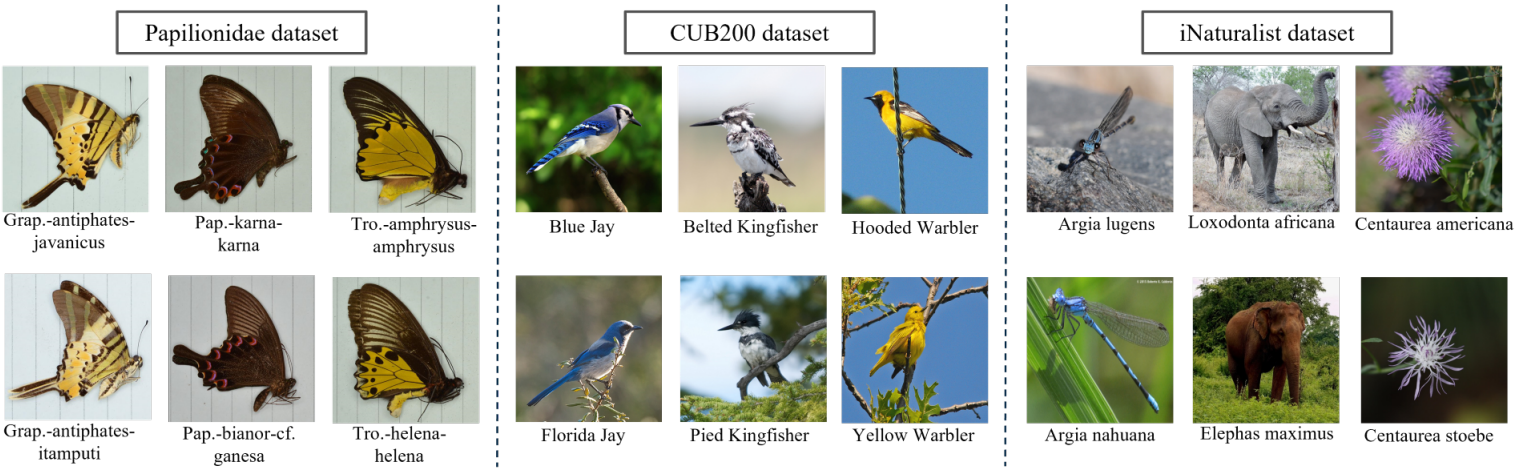


FIGURE 4.2: Examples showing the fine-grainedness nature of the three datasets.

4.2.3 Constructing In-Distribution and Out-of-Distribution Datasets

4.2.3.1 Datasets for Near-OOD

For testing the 'near-OOD novel class detection' (or near-OOD detection in short) case, the three datasets are split into ID and OOD. Going by the idea that "there is always more to know than what you already know", the number of ID classes is chosen to be less than half of the dataset.

Formulating the split of a dataset into ID and OOD Consider a dataset D_{org} having C_{org} number of classes and a total of N_{org} images. The dataset is split into ID and OOD datasets- D_{id} andD_{ood}, with C_{id} and C_{ood} number of classes respectively. The set of classes C_{id} is randomly chosen without replacement from the set C_{org} using a uniform distribution. The total number of images in D_{id} is N_{id} and D_{ood} is N_{ood}. After the split of the dataset, the following heuristic requirements are checked:

- The number of images in D_{id}, i.e., N_{id} must not deviate by more than 10% of $N_{org} * (\frac{C_{id}}{C_{org}})$. This is to ensure that all the classes having a large number of images do not entirely belong to ID or OOD split.
- Class imbalance factor: The class imbalance factor for the D_{id} is similar to or higher than that of D_{ood}. This is to ensure that the 'long-tailed' distribution characteristics of the original dataset are maintained in the ID dataset.
- If a subset of classes in C_{org} have similar concepts or are closely related to form a set S, then it is made sure that not all of them are in D_{id}. For instance, in the iNaturalist dataset, a particular set of 77 classes can further be grouped into a so-called 'super category' called Animalia. It is made sure that not all the 77 classes in Animalia are in the D_{id} dataset after the split.

If the heuristics are not met after splitting a dataset, a few new attempts with a different seed number are tried until the heuristics are met. Although not encountered, if these attempts do not give ID and OOD datasets with the stated requirements, then manually rearranging the least number of classes to meet the requirements was done.

The ID and OOD dataset splits are named using the following syntax: the original dataset name followed by the number of classes in the corresponding splits. For instance, the ID dataset made from the Papilionidae dataset consists of 50 classes and hence is referred to as Papilionidae50 (or Pap.50 in short).

The dataset D_{id} will be split into training, validation, and test sets using a stratified split at a ratio of 80:10:10. It is made sure that each class has at least one image in these splits as well. Table 4.1 shows the characteristics of the three datasets before and after splitting them into ID and OOD.

Dataset	Taxonomic Hierarchy	Split	Number of classes	Number of images	Imbalance factor
Papilionidae	Butterfly family 'Papilionidae'	Original	112	8243	0.3057
		ID: Pap.50	50	3489	0.4195
		OOD: Pap.62	62	4754	0.3137
CUB200	Class 'Aves'	Original	200	11788	0.0002
		ID: CUB80	80	4737	0.0002
		OOD: CUB120	120	7051	0.0004
iNatrualist	Whole domain	Original	5089	675170	0.3274
		ID: iNat.2100	2100	284419	0.0957
		OOD: iNat.2989	2989	390751	0.0905

TABLE 4.1: Constructing the ID and near-OOD datasets from the Papilionidae, CUB-200 and iNatrualist datasets.

4.2.3.2 Datasets for Intermediate- and Far-OOD

For testing the OOD detection for 'intermediate-OOD' and 'far-OOD' cases, the following datasets are used as OOD datasets.

- Costa Rica Moths dataset: For testing the intermediate-OOD detection case on the Papilionidae dataset, the Costa Rica Moths dataset [75] is used. The dataset contains 2310 images of moths taken similar to the images in the Papilionidae dataset. The right half of the images are removed to make them more similar to the images in the Papilionidae dataset.

- Places365: Places365 [76] is a large-scale scene recognition dataset created for developing algorithms for scene recognition. The dataset contains 365 scene categories, such as beaches, streets, and forests, among others. In this work, the validation split of the first version of the dataset containing 36500 images is used. There are no overlapping classes between the Places365 and the ID datasets considered. However, images from biodiversity-related classes such as 'rain forest', and 'bamboo forest' in the Places365 dataset could be challenging to be classified as OOD images when the ID dataset is CUB80 or iNaturalist2100. In CUB80 and the iNaturalist2100 datasets,

a majority of the subjects (birds, animals) have 'forest' backgrounds. Hence for these datasets, the Places365 dataset is considered as 'intermediate-OOD', while for the Papilionidae50 dataset, it is considered as 'far-OOD'.

- Describable Texture Dataset: Describable Texture Dataset (DTD) [77] is a collection of textural images that covers a wide range of textures and patterns. The dataset contains diverse textures and patterns from natural and manmade objects. There are no overlaying concepts between the DTD dataset and the ID datasets and hence is used as a 'far-OOD' case. Using the DTD as an OOD dataset provides a useful evaluation of the generalisation ability of OOD Detection methods. The textures and patterns in the DTD dataset are significantly different from those in the biodiversity dataset and testing a model on these textures can provide insight into the robustness of the methods to detect completely unseen patterns and textures.

- Random pixel images: As done in [3] and [37], a synthetic Gaussian noise dataset and synthetic uniform noise dataset are also used as OOD Datasets. The Gaussian noise dataset consists of 10,000 random 2D Gaussian noise images, where each RGB value of every pixel is sampled from a Gaussian distribution with mean 0.5 and unit variance. Similarly, the synthetic uniform noise dataset consists of 10,000 images where each RGB value of every pixel is independently and identically sampled from a uniform distribution on [0, 1]. These sets of images are not biased towards any particular image class from the ID datasets and represent the set of OOD images from which no relevant features or patterns can be extracted. The main set of results is presented on the Gaussian noise dataset.

- ChestXRay2017: The ChestXRay2017 [78] is a medical imaging dataset that comprises 5857 chest X-ray images. The 5233 images from the training set of the dataset are used as the OOD dataset. Although x-ray images do not usually appear in a non-medical biodiversity setting, the dataset is the least similar OOD dataset to the other ID datasets and hence is used as a set of extreme OOD example images in this work.

The set of OOD datasets and their domain similarity scores with respect to each ID dataset is shown in Figure 4.4.

4.2.3.3 Datasets for Anomaly Detection

This subsection explains the construction of datasets used for testing the anomaly detection capability of the methods.

By applying 7 different types of corruptions, anomalies are added to the images in the test set of each of the ID datasets (expect iNaturalist2100). For each ID test set, there would be 7 corresponding anomalous (OOD) test sets. The following corruptions that are more likely to appear while capturing images in a biodiversity environment are chosen:

- Camera-related corruptions
  - **ISO noise** ISO noise is the random variation in brightness and/or colour information in images usually caused by bad lighting conditions or low-quality camera sensors

- Image capturing related corruptions
  Some of the common issues that occur while capturing images in the wild are covered under this category:

- **Motion Blur** Motion blur results when an image is captured when the camera, the subject, or both are in motion.
- **Defocus** Defocused blur occurs when the subject is not in focus.
- **Coarse dropout** In this corruption, random rectangular sections of the image are dropped or blacked out. Coarse dropout on images to mimic occlusions that obscure the subject partially or fully.

- Weather-related corruptions
  The following weather conditions that visually obstruct the subjects are chosen:
  - **Fog**
  - **Rain**
  - **Snow**

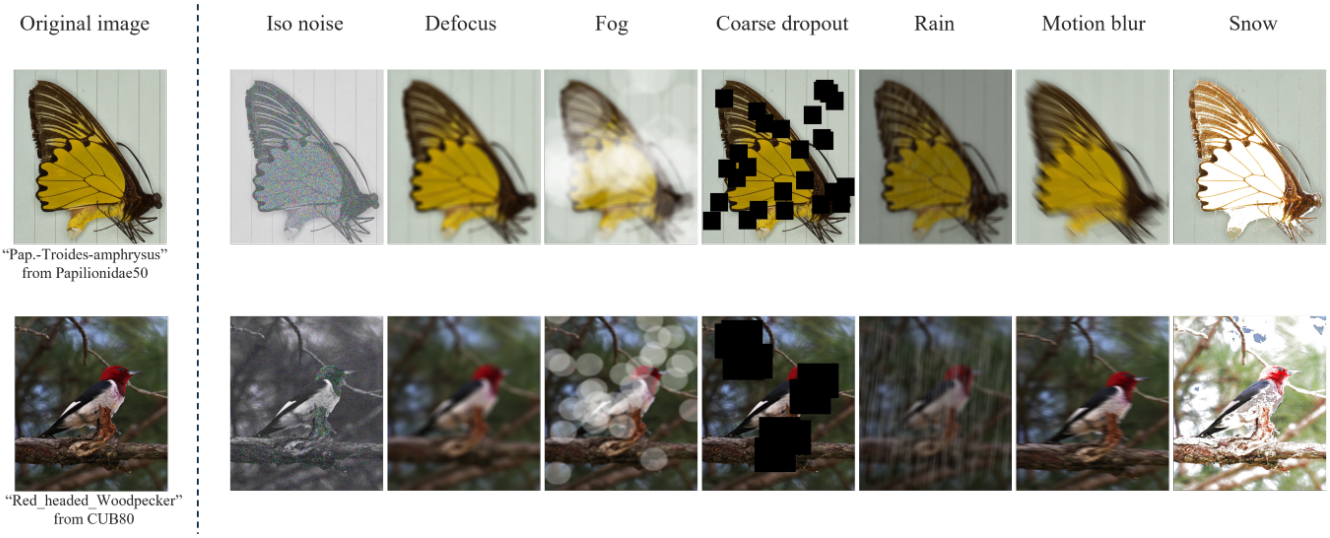


FIGURE 4.3: Examples of images from the Papilionidae50 and CUB80 dataset with the seven types of corruptions applied to make them anomalous

The strength of the corruptions is tuned until the test set has a classification accuracy of 50%($\pm$0.05), i.e., half of the test set is classified incorrectly. This was done to standardise the amount of corruption applied. Anomaly detection is tested only on the relatively smaller Papilionidae50 and CUB80 datasets. This covers the case of anomaly detection on long-tailed and balanced datasets respectively. Anomaly detection is not tested on the iNaturalist2100 dataset as applying each type of corruption to get 50% accuracy on the test set of the iNaturalist2100 dataset would be highly time-consuming and storing those corrupted images' datasets requires more than 1000 GB of disk space.

Figure 4.3 shows examples of these corruptions applied to the example images from CUB80 and the Papilionidae50 dataset.

4.3 Experiment Setup

4.3.1 Training the Convolutional Neural Network Classifier

4.3.1.1 Choice of Convolutional Neural Network Architecture

For training the classifier on the ID datasets, EfficientNetV2 [70] architecture was chosen over other architectures due to its performance-to-parameters ratio. The use of OOD detection for biodiversity is often in the wild where the neural networks run on low-power embedded systems. Hence, there is a need for using models which require lower computing power as well as time. EfficientNetV2 has better parameter efficiency and can achieve the same or better accuracies with fewer parameters when compared with ResNet-RS [79] or ViT [80] models [70]. Furthermore, EfficientNetV2 uses Fused-MBConv [81] blocks that can utilise power-efficient hardware accelerators such as Edge TPU [82] which are often used to infer models on low-power devices. Finally, EfficientNetV2 models can be trained relatively faster. This is especially useful while training on large datasets like the iNaturalist dataset. From the family of EfficientNetV2 models, the EfficientNetV2-M variant is chosen as it can achieve similar accuracies as the EfficientNetV2-L (85.1% vs 85.7% on ImageNet-1K [70]) using roughly half the number of parameters and floating point operations (FLOPs).

4.3.1.2 Training Setup and Parameters

An EfficientNetV2-M model with weights pre-trained on the ImageNet-1K [8] dataset is trained on each of the in-distribution (ID) datasets. As mentioned earlier, 80% of each ID dataset is used for training the corresponding model. The input size has been set to 480x480x3. Three basic data augmentations - *RandomHorizantalFlip* [83], *RandomVerticalFlip* [84] and *RandomRotation* [85] provided by the Torchvision [86] library are applied to the input images during the training. The *RandomRotation* augmentation was applied with a maximum rotation range of 45°. All three augmentations are applied with a probability of 0.5. Stochastic gradient descent with a momentum of 0.9 has been used as the training optimiser. Table 4.2 shows the hyperparameters used for training as well as the accuracy (number of images correctly classified/total number of images) achieved on each dataset's test set.

	Papilionidae50	**CUB80**	**iNaturalist2100**
Initial learning rate	0.001	0.010	0.010
Learning rate scheduler	CosineAnnealingWarmRestarts [87],[88]	ExponentialLR [89]	ExponentialLR [89]
Batch size	16	16	16
Number of epochs	24	14	35
Accuracy	95.7	94.54	84.98

TABLE 4.2: Hyperparameters used for training the EfficientNetV2-M model on the three in-distribution datasets. The accuracy stated is on the respective test sets.

4.3.2 Hardware and Software Setup

All the experiments in this work are performed on a system with the following properties:

- CPU: Intel Xeon Gold 6330 CPU @ 2.00GHz
- GPU: NVIDIA GeForce RTX 3090 with 24 Gigabytes of GPU memory

- Python version: 3.9.13
- PyTorch version: 1.11
- CUDA version: 11.2

For PyTorch [90], the *torch.backends.cudnn.deterministic* and *torch.backends.cudnn.benchmark* flags are set to *true*. The random seed number is set to *2511* wherever it is possible to do so.

Table 4.3 lists the important libraries and the open-source code used in this work.

Method	**Source**	**Comments**
Baseline MSP	–	–
Energy	–	–
ODIN	[91]	Adapted from official implementation
GradNorm	[91]	Official implementation
Global PCA FRE	–	–
DkNN	adapted from [92]	Official implementation
EnWeDi	–	–
SLOF	–	–
Functionality	**Library**	**Function names**
Adding corruptions to create anomaly datasets	Albumentations 1.3.0 [93]	CourseDropout(), Defocus(), Fog(), IsoNoise(), MotionBlur(), Rain(), Snow()
EMD for domain similarity score	pyemd 1.0.0 [94]	emd()
PCA for FRE methods	scikit-learn 1.1.1 [95]	decomposition.PCA()
kNN search for kNN-based methods	Faiss 1.7.2 [44]	GpuIndexFlatL2() , IndexFlatL2()
Entropy calculation in EnWeDi	scipy 1.8.1 [96]	stats.entropy()

TABLE 4.3: List of open-source code and the libraries (and functions) used in this work

4.4 Extracting Feature Embeddings

Most of the feature embedding-based methods mentioned in the literature section (Section 2.3) make use of the feature embeddings from the penultimate layer of a convolutional neural network (CNN). This is because, in a CNN, the deeper layers i.e., layers closest to the output, typically extract more abstract and high-level features from the input data. Shallower layers extract simpler and low-level features like edges, corners, etc. Hence, the feature embeddings extracted from the last layers are more specific to the dataset that the CNN has been trained on. In this work, all feature-based methods are tested with two sets of feature embeddings:

- **Penultimate layer's feature embeddings** Penultimate layer feature embeddings obtained after the global average pooling layer from Layer 8 of EfficientNetV2-M [70]. Each feature embedding is of length 1280.
- **Stacked feature embeddings** Feature embeddings extracted using the global average pooling layer from all the major intermediate layers – Layer 0 to Layer 8 and stacked one after the other to make a single feature vector of length 2608. Such a feature embedding would represent both general as well as high-level features of the

image. Table A.2 in the Appendix gives the list of layers from which the features are extracted and their respective lengths.

4.5 List of Experiments

RQ1: Measurement of OOD detection difficulty The domain similarity score (Equation 4.3) is calculated between each combination of ID and OOD datasets. The details are mentioned in Section 4.1. The scores are shown in Figure 4.4 and will also be indicated while presenting the corresponding results.

RQ2: Improvements to existing methods The experiments made for answering RQ3 and RQ4 are used to find the limitations of the existing methods. Why the proposed method - EnWeDi (explained in Section 3.2.8) performs better than the existing state-of-the-art OOD detection method is discussed in Section 6.4.

RQ3: Effect of stacking The four feature embedding-based methods (Global PCA FRE, DkNN, EnWeDi and SLOF) have been tested with feature-embeddings extracted from intermediate layers (as mentioned in Section 4.4). This was done on the near-OOD detection case on the Papilionidae50 and CUB80 datasets.

RQ4: OOD detection methods' performance on a range of OOD detection difficulties All the shortlisted methods (marked in bold in Table 3.1) are tested on all the three datasets - Papilionidae50 (Pap.50), CUB80 and iNaturalist2100 (iNat.2100) against datasets corresponding to near-, intermediate- and far-OOD. Due to the large number of images in iNaturalist2100 (0.3 million), the task of anomaly detection is tested only on the Papilionidae50 and CUB80 datasets. All the combinations of the ID and OOD datasets for each OOD detection task are shown in Figure 4.4.

Each method is tested at various hyperparameters (as described in Subsection 4.5.2). The evaluation metrics are explained in Section 4.6.

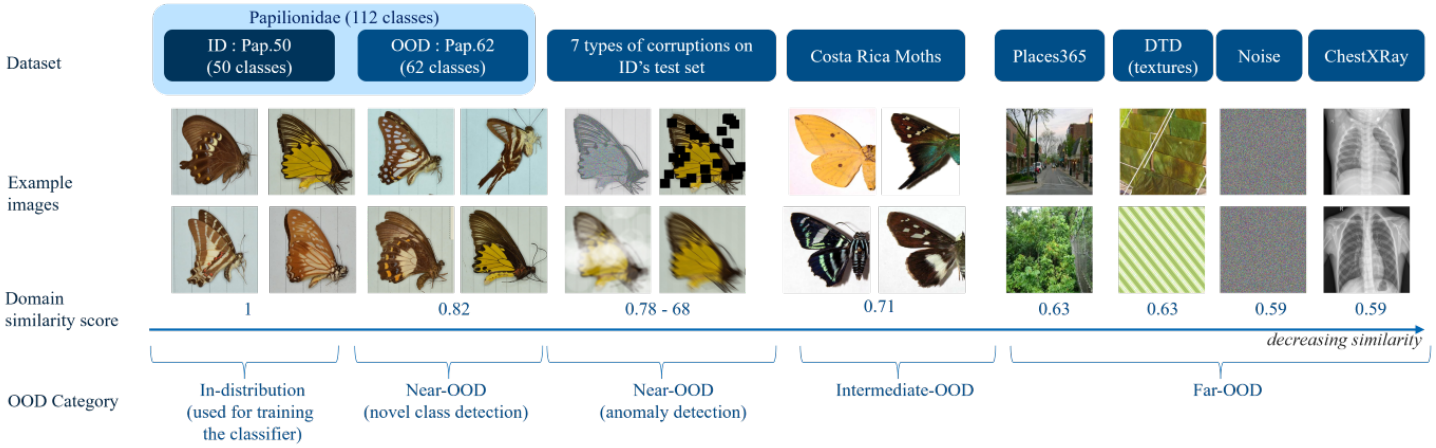
4.5.1 Effect of the Amount of In-Distribution Data Used on Feature Embeddings-Based Methods

The feature embeddings-based methods considered in this work require a certain amount of in-distribution data for either referencing or learning from it. For instance, the PCA-based FRE methods require feature embeddings from the ID data to 'train' the PCA model. Similarly, for kNN-based methods, the ID dataset images' feature embeddings are required as 'reference' or 'training' data for the kNN search.

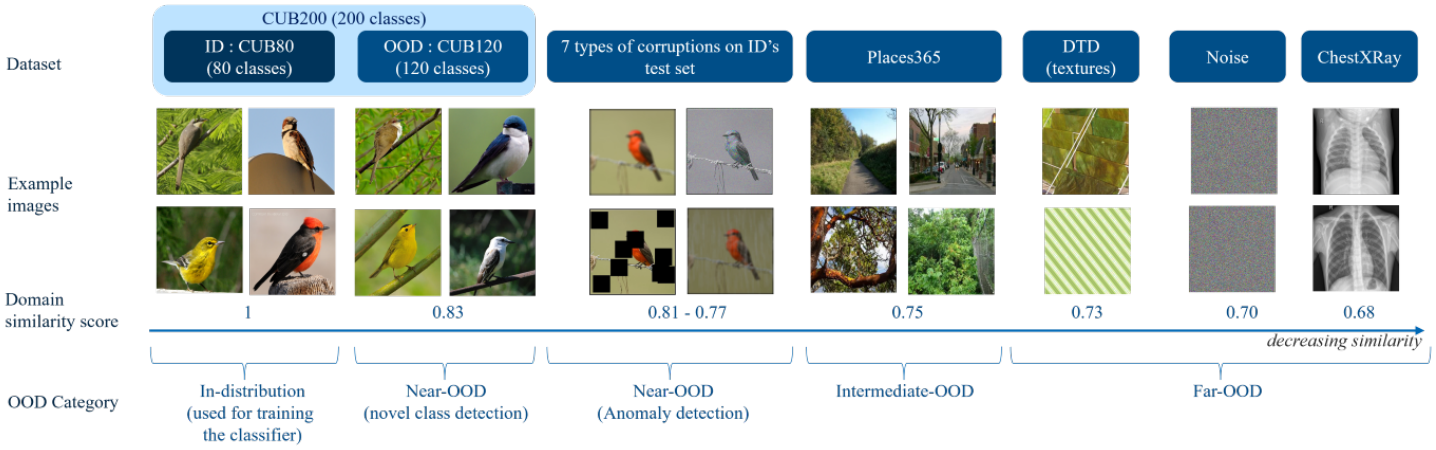
To estimate the performance of feature embedding-based methods on smaller ID datasets, the methods are also evaluated with subsets of ID datasets. To create such a subset from an ID dataset, a fixed fraction of the images from each class is selected (stratified selection). The set of fractions is uniformly spaced between 0.1 (10% of each class) to 1.0 (all the images in the class - whole dataset).

4.5.2 Setting Hyperparameters for OOD Detection Methods

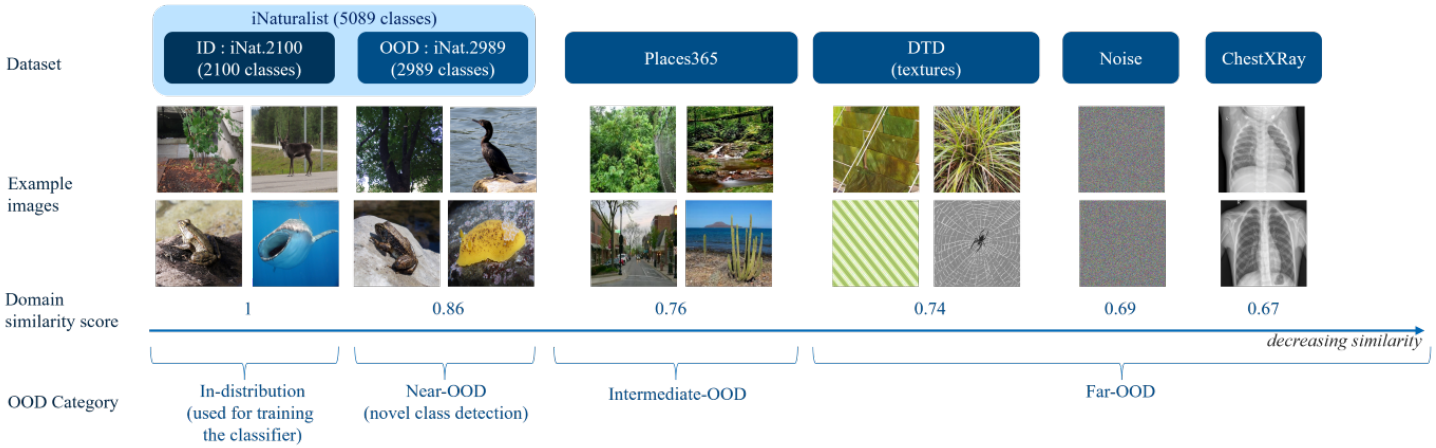
1. **Baseline Methods:** The methods are evaluated on metrics that do not require implicitly setting the OOD decision threshold. Hence, the Baseline MSP method does not have any hyperparameters to set.



(A) Papilionidae50 as in-distribution dataset



(B) CUB80 as in-distribution dataset



(C) iNaturalist2100 as in-distribution dataset

FIGURE 4.4: ID and OOD datasets on which the OOD detection methods are tested. The domain similarity score (Equation 4.3) is between the OOD dataset and the ID test set

2. **Energy:** The Energy method takes the *temperature* (T) hyperparameter for performing temperature scaling. T has to be ≥ 1 for the method to be effective [2]. The authors in the literature on the Energy method perform an ablation study with T ranging from 1 to 1000 and suggest that the highest accuracy can be achieved by setting $T = 1$. In this research, the energy method was tested with $T = \{0.1, 1, 10, 100, 1000\}$

3. **ODIN:** The ODIN method has two hyperparameters - the *temperature* (T) and the ϵ (perturbation magnitude). Similar to the Energy method, the hyperparameter T is used for temperature scaling in ODIN and was set to the same values as mentioned above. The literature of ODIN method [3], suggests experimenting with perturbation magnitudes (ϵ) between 0.001 to 0.004. In this work, it was set to $\{0.001, 0.002, 0.01, 0.1\}$.

4. **GradNorm:** GradNorm method also uses the *temperature* (T) parameter for temperature scaling. From the literature of GradNorm [4], "T = 1 is optimal, while either increasing or decreasing the temperature will degrade the performance". However, for a fair comparison, GradNorm was evaluated with temperatures $T = \{0.1, 1, 10, 100, 1000\}$ (same as Energy and ODIN). Additionally, from the ablation studies done by authors of GradNorm, the following hyperparameters are set:

   - Applying GradNorm applied at the last layer gives higher accuracies than when applied on shallow layers (layers near input). Hence, GradNorm is applied only at the last layer in this work. Accordingly, the weights from the final fully connected layer of the CNN are considered as the set of parameters w in the Equation 3.7.
   - Using L_1 norm in GradNorm OOD score calculation (Equation 3.7) gives better results than any other L_p norm. Hence, in this work, the L_1 norm is used in all experiments of GradNorm.

5. **Global PCA FRE Method:** For PCA, the fraction of the variance (σ) of the training set's feature embeddings that should be retained in the reduced feature embeddings is the hyperparameter. The literature of the Global PCA FRE method [5] does not mention any ablation studies or the value of σ used for the results. Hence, in this research, σ was set with uniformly spaced values from 0.9 to 0.99. Along with this, the method was also tested on these extreme values: {0.6, 0.8} (low variance retention) and {0.995, 0.999} (high variance retention).

   The same list of hyperparameters was used for the Per-class PCA FRE method when experimenting on the Papilionidae50 and CUB80 datasets. For the iNaturalist2100, the Per-class PCA method takes more than 90 hours to test on all the OOD datasets. Hence, the Per-class PCA FRE method was tested with $\sigma = \{0.8, 0.9\ 0.95, 0.98\}$

6. **DkNN:** For DkNN, the value of k used for the kNN search is the hyperparameter. In the corresponding literature [6], the method was evaluated with 10 different values of k ranging from 1 to 5000. From the results presented in the same literature, it is observed that setting k greater than the number of images per class in the balanced ID dataset will not improve the OOD detection accuracy. However, in this research, except for the CUB80 dataset, the other ID datasets are not balanced in terms of the number of images per class. Hence, the value of k was set as following:

- For the Papilionidae50 dataset, the average number of images per class is 62. Hence, k is set from 1 to 62 (both inclusive). Additionally, 20 more values of k are sparsely selected from the range of 63 to 450.
- For the CUB80 dataset, the value of k is set with uniformly spaced values from 1 to 50 (the average number of images per class in CUB80 is 47). Furthermore, 20 more k values ranging between 60 and 450 are chosen.
- For the iNaturalist2100 dataset, the value of k is set with uniformly spaced values from 1 to 50. Additionally, 10 more values of k from the range of 50 to 2000 are chosen.

7. **EnWeDi:** Since the EnWeDi method is proposed as an improvement to the DkNN method, for a fair comparison, the values of k used for DkNN are used for EnWeDi as well.

8. **SLOF:** SLOF method was experimented with the same list of k values used for DkNN and EnWeDi.

4.6 Evaluation

All the out-of-distribution (OOD) detection problems in this work are treated as binary classification problems. Following the literature, the in-distribution images are considered positive examples and the OOD images are considered negative examples throughout this thesis.

For most of the OOD detection applications, the objective is usually to prevent OOD images from getting classified as belonging to ID (to reduce the open space risk). For instance, in autonomous driving, the objective is not to identify the maximum number of unknown objects as unknown objects. Instead, the objective is to prevent an unknown object from being a known object. Therefore, it is apt to consider OOD images as negative examples.

4.6.1 Threshold Independent Metric - Area Under the Receiver Operating Characteristics Curve

To classify whether an input image is in-distribution or out-of-distribution, the OOD decision function (Equation 2.1) requires a threshold on the OOD score. However, specifying a threshold depends upon the trade-offs between the number of acceptable false negatives (fn) and false positives (fp) and can vary from application to application. Hence, in this work, the area under the receiver operating characteristic curve (AUROC), a threshold-independent performance evaluation [97] metric is used. The ROC curve is a graph showing the true positive rate ($TPR = tp/(tp+fn)$) and the false positive rate ($fpr = fp/(fp+tn)$) and indicates the performance of a classification model at various possible classification thresholds. For a binary OOD detection problem, the AUROC value can be interpreted as the probability that an ID image (treated as positive) has a greater detection score than an OOD image [98]. Hence, a random guess OOD detection method would have an AUROC value of 0.5, while a perfect OOD detection method would have an AUROC of 1 (or 100%).

4.6.2 Threshold Dependent Evaluation Metrics

In a few applications, the OOD detector is required to meet certain operation requirements in terms of true positive rate (TPR) at a fixed false positive rate (FPR) or vice-versa. Two

such metrics - FPR at N% TPR and TPR at N% FPR are also used to evaluate the OOD detection methods in this research. Unlike AUROC, these metrics represent the performance of the OOD detection methods at a fixed threshold. These metrics can be especially useful in cases where the OOD detection methods achieve a very high and/or similar AUROC value.

False Positive Rate at N% True Positive Rate False positive rate (FPR) at N% true positive rate (TPR) or FPR@TPRN is an important evaluation metric for OOD detection as it measures the ability of an OOD detection method to not classify OOD images as belonging to ID while maintaining a certain level of TPR (classifying ID images as ID). A lower FPR@TPRN value indicates better performance. Lower FPR@TPRN implies that there is a smaller chance of OOD images triggering false alarms (OOD images getting classified as ID images) when the OOD method can classify N% of the ID images correctly. This metric is useful when the objective is to have a set fraction (N%) of the ID samples to be classified as ID.

True Positive Rate at N% False Positive Rate True positive rate (TPR) at N% false positive rate (FPR) or TPR@FPRN is useful in an application when the acceptable false positive rate can not exceed a certain value. In such cases, the decision threshold is set to a value such that the OOD detection method does not classify more than N% of OOD images as ID. For instance, consider an image classification model deployed to identify the species of a crop plant in an agriculture field. In such cases, misclassifying an invasive plant species (OOD) as a crop plant can lead to potentially harmful effects on the crops. In such cases, the false positive rate (invasive species getting classified as crop species) is fixed at a low value (such as 5%).

4.6.3 Choice of Evaluation Metric Based on OOD Detection Difficulty

Near-OOD detection: Out of all the OOD detection tasks considered in this work, the near-OOD detection task is the most difficult. Hence, the methods are evaluated on all three evaluation metrics - AUROC, TPR@FPRN and FPR@TPRN. Following the literature, for the TPR@FPRN and FPR@TPRN, the methods are compared at the requirements of TPR@FPR5 and FPR@TPR95. These requirements strike a reasonable trade-off between finding OOD images and classifying ID images. Consider the above-mentioned crop-plant scenario where it can be difficult to differentiate between a weed plant and a crop plant (near-OOD). While attaining a 1 % FPR or 99% TPR could be too 'uncompromising', having 10% FPR or only 90% TPR could be too 'forgiving'. Hence a 5% FPR or a 95% TPR could serve as attainable yet realistic requirements.

Intermediate- and far-OOD detection: For the relatively easier OOD detection tasks, such as intermediate- and far-OOD detection tasks, TPR@FPR5 and FPR@TPR95 could be too lenient. Once again, consider the above-mentioned crop-plant scenario. In such cases, it is not crucial to identify objects from far-OOD such as farm animals as OOD. However, from a user's perspective, a farm animal must not be classified as a crop plant, i.e., a low false positive rate is preferred. However, this is (supposed to be) relatively easier than the near-OOD task and many OOD detection methods can attain a high TPR at requirements such as 5% FPR. Therefore, the results of the methods are compared at a stricter requirement of TPR@FPR1.

Anomaly detection: In anomaly detection applications, anomalous images getting classified as non-anomalous must be prevented as much as possible. Hence, more often than not, a predetermined maximum false positive rate is set and the anomaly detection capability of the methods is measured under this specific threshold for false positives. The goal is then to classify as many true positives (non-anomalous images) as positives at the set false positive rate. Therefore the results for anomaly detection are presented with the evaluation metric - true positive rate at a 1% false positive rate (TPR@FPR1).

4.7 Inference Time Measurement

The inference time of the OOD detection methods (and their variants) is measured on the three ID datasets. Inference time was measured from the time the input image is available in memory until the OOD detection score was generated by the methods. For the score-based methods, the inference time includes the time taken for the forward propagation of the CNN model. Similarly, for the feature embeddings-based methods, the time taken to extract the features(from the penultimate layer) is included in the inference time.

The inference time of each OOD detection method is measured on a test set of 10,000 OOD images and the average inference time is calculated for each method. To avoid the influence of the time taken for GPU warm-up, the inference time measured on the first test image is discarded.

For PCA-based methods, the fraction of the original data's variance to be retained was set to 95%. For methods that require a kNN search, k was set to 100. Faiss [44] can be configured to take advantage of the parallelisation capabilities of a GPU for a (presumably) faster kNN search. Hence, the inference time of the kNN-based methods is measured with two configurations - kNN search using CPU and kNN search using GPU. For the other methods, the hyperparameters do not have any influence on the inference time.

Chapter 5

Results

The results of near-out-of-distribution (near-OOD) detection, anomaly detection and intermediate/ far-OOD detection of selected methods are presented in Sections 5.1, 5.2 and 5.3. The effect of the amount of reference ID data on the OOD detection accuracy for feature embeddings-based methods is shown in Section 5.4. In Section 5.5, the inference time of OOD detection methods is presented. Section 5.6 shows the results of applying kNN- and PCA FRE-based methods per class instead of applying them globally. The results with feature embeddings stacked are shown in Section 5.7.

Choice of hyperparameters For an OOD detection method, operating on a fixed hyperparameter might not give the best results across all the OOD detection tasks. However, once deployed in an open world, a particular method needs to perform out-of-distribution detection for any type of OOD. Changing hyperparameters on-the-fly for each type of OOD is not practical as it would require predicting the type of OOD an input image belongs to.

The near-OOD task is often more frequently encountered by a real-world classification application than other tasks. Hence, the results presented for all the tasks (near-, intermediate/far-OOD and anomaly detection) are obtained by using the hyperparameters that gave the highest AUROC on the near-OOD tasks for the respective datasets (hyperparameter optimisation). One exception to this is the case of evaluating the near-OOD detection with a fixed operating TPR or FPR requirement (Subsection 5.1.2). In such cases, since the OOD detection task needs to achieve a certain requirement, the methods are compared at hyperparameters that achieve the best possible FPR@TPR95 or TPR@FPR5.

5.1 Near Out-of-Distribution

In Subsection 5.1.1, the methods are compared in terms of area under the receiver operating characteristic curve (AUROC) - a threshold-independent metric. In Subsection 5.1.2 the methods are compared against a specific set of acceptable false positive rates and required true positive rates.

5.1.1 Threshold Independent Accuracy Performance - AUROC

5.1.1.1 Papilionidae50 Dataset

The AUROC values of the 8 OOD detection methods are shown in Figure 5.1. The proposed method - Entropy weighted nearest neighbour's distance (EnWeDi) achieves the highest

AUROC value outperforming the DkNN method by more than 5%. More often than not, the feature embeddings-based methods perform better than the score-based methods (MSP, Energy, ODIN, and GradNorm). With an AUROC value of 84.81%, the maximum softmax probability (MSP) baseline method is already helpful in distinguishing between ID and OOD images and is not the worst performing method.

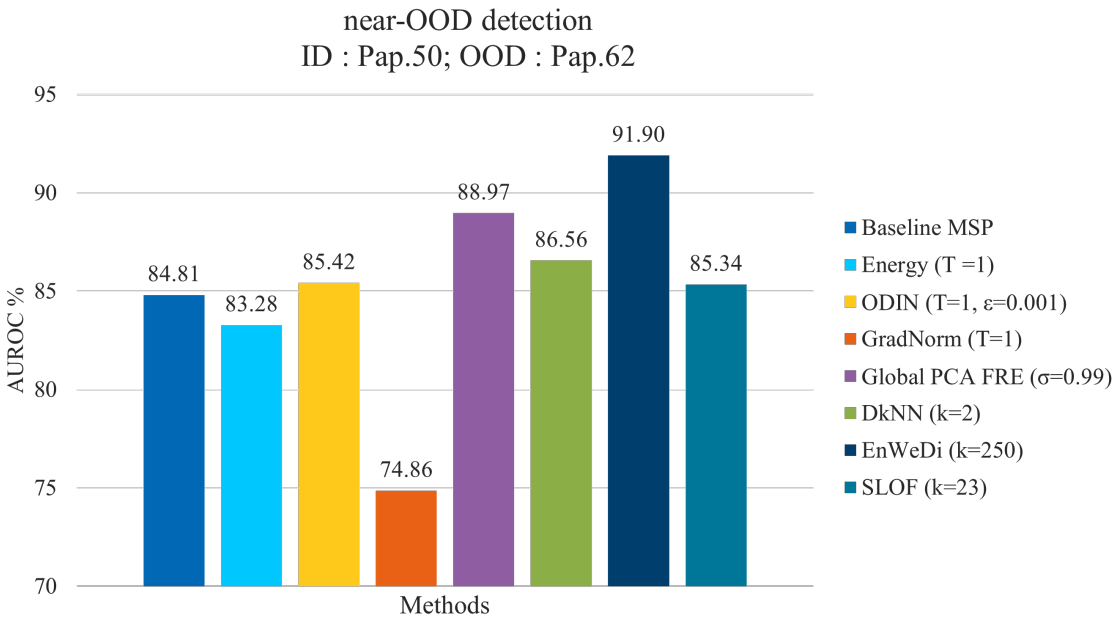


FIGURE 5.1: Results on the near-OOD detection on the Papilionidae50 dataset.

5.1.1.2 CUB80 Dataset

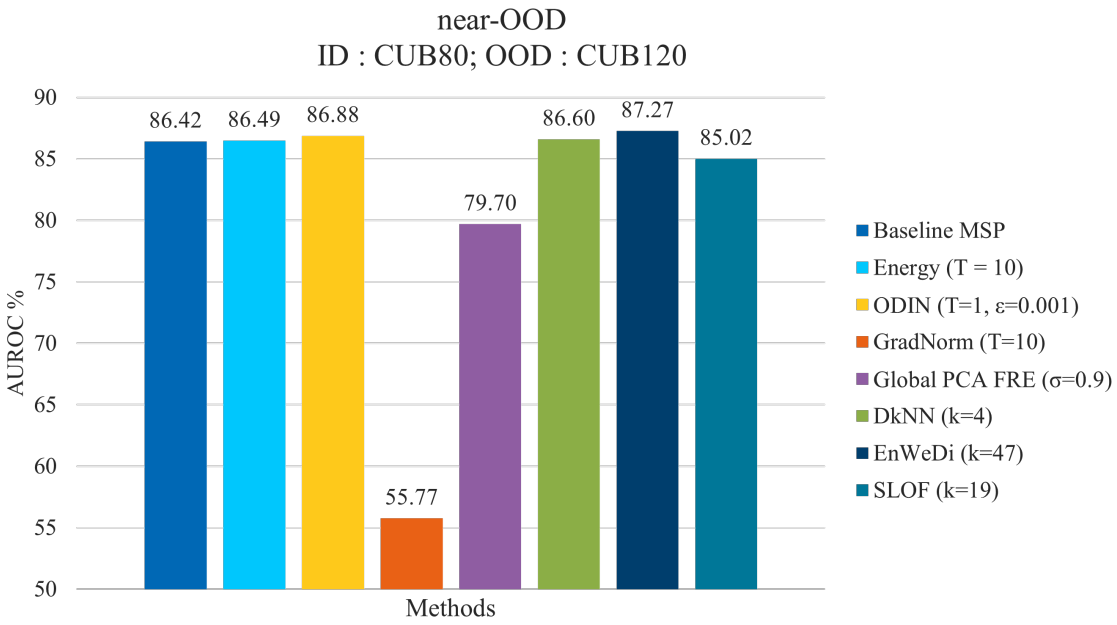


FIGURE 5.2: Results on the near-OOD detection on the CUB80 dataset.

On the CUB80 dataset, as shown in Figure 5.2, using the AUROC metric for evaluating the methods does not reveal much difference between the near-OOD detection capabilities of the methods. The AUROC values for 6 out of 8 methods are within the range of 85.0% to 87.5% giving very little insight into which method performs the best on the well-balanced CUB80 dataset for the near-OOD detection problem. Hence, as it will be shown in the next subsection, evaluating the methods at a fixed threshold value is more useful in this case.

However, a few interesting observations can be drawn from the results. The Global PCA FRE method, which was the second best method for near-OOD detection on the Papilionidae50 dataset, does not show a similar capability on the CUB80 dataset. It can also be seen that with an AUROC of 55.77 %, the GradNorm method performs the worst on the CUB80 dataset, almost similar to a random ID vs OOD classifier.

5.1.1.3 On the iNaturalist2100 Dataset

Testing the near-OOD detection task, on the large long-tailed distribution dataset iNaturalist2100 breaks a few of the patterns observed in the results on the above two datasets. ODIN method achieves the highest AUROC, although it outperforms the EnWeDi method by a small margin of 0.3%. It is to be noted that ODIN was the best-performing method for the Papilionidae50 and CUB80 datasets in the set of output scores-based methods. The Energy method shows similar OOD detection performance as the Baseline MSP method. An interesting result was how poorly the Global PCA FRE, a feature-based method, performed in this case. It can also be observed that the DkNN and EnWeDi methods are in the top 3 methods (in terms of AUROC) for all three datasets, with EnWeDi performing better than DkNN in all the cases.

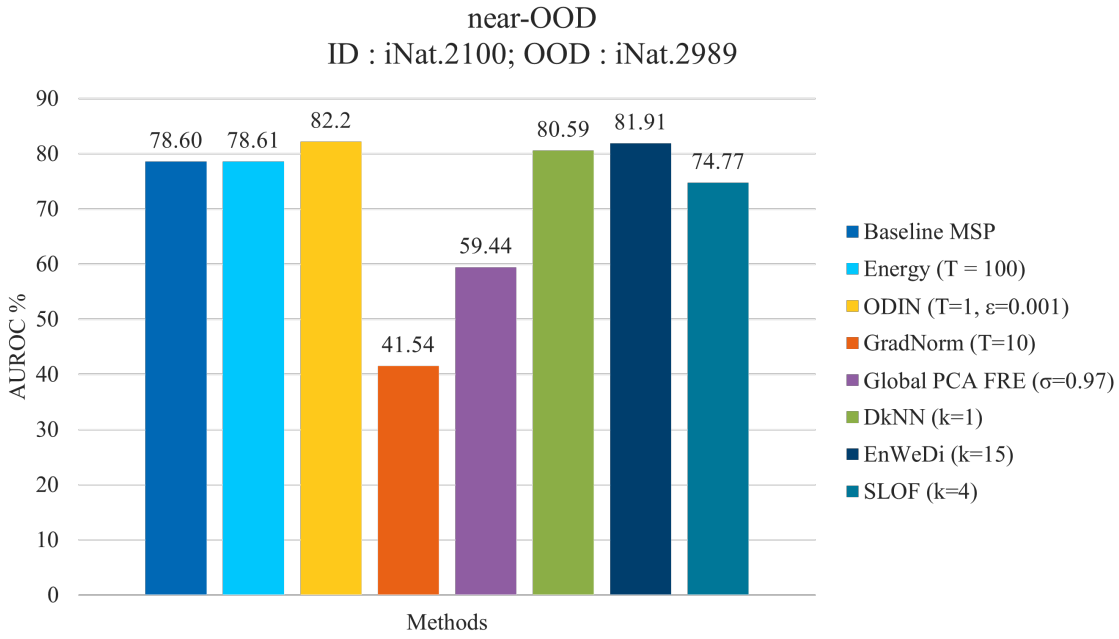


FIGURE 5.3: Results on the near-OOD detection on the iNaturalist2100 dataset.

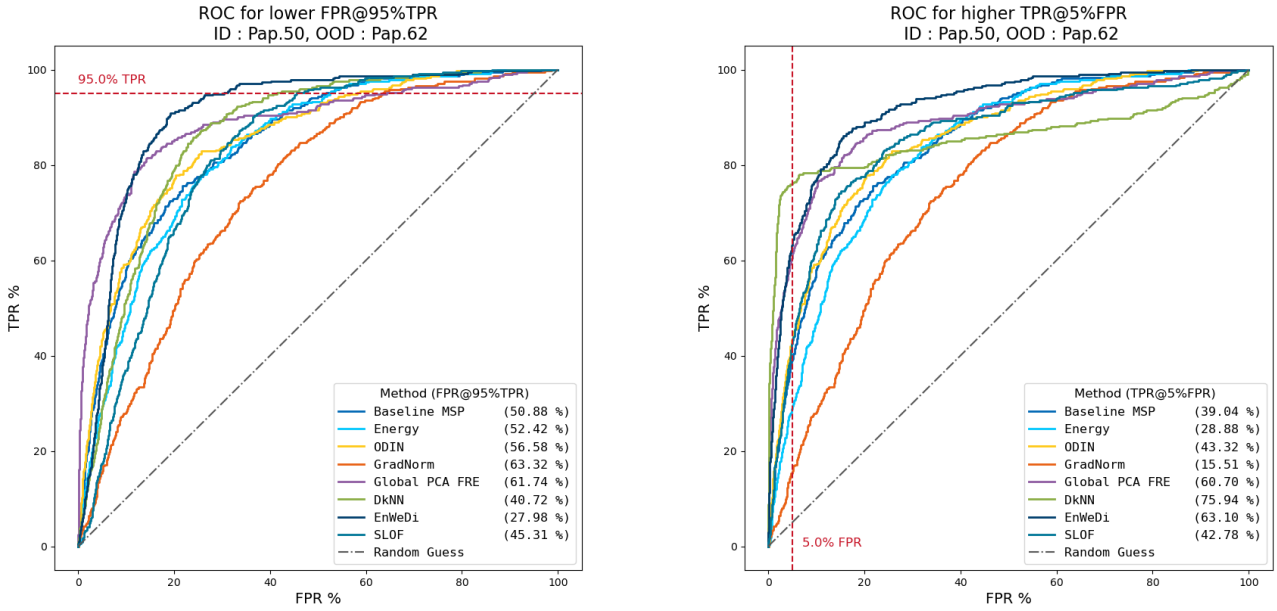
5.1.2 OOD Detection Performance on Specific Operation Requirements

As mentioned earlier, **in this subsection, the results shown are obtained using the hyperparameters that give the lowest FPR@TPR95 and highest TPR@FPR5 values for each method.**

5.1.2.1 Papilionidae50 Dataset

Figures 5.4a and 5.4b show the ROC curves for achieving the lowest FPR@TPR95 and highest TPR@FPR5 possible by each method respectively.

The proposed method - EnWeDi achieves the best (lowest) FPR@TPR95 with a significant margin of more than 12% from the next best method (DkNN). With a very high FPR@TPR95 of 61.74%, the Global PCA FRE method which achieved the second highest AUROC in this OOD detection task does worse than the Baseline MSP method (50.88%). It is also observed that all 8 methods can achieve a significantly lower FPR@TPR95 when compared to a random guess classifier which would have a 95% FPR@TPR95.



(A) FPR@TPR95 (lower is better)

(B) TPR@FPR5 (higher is better)

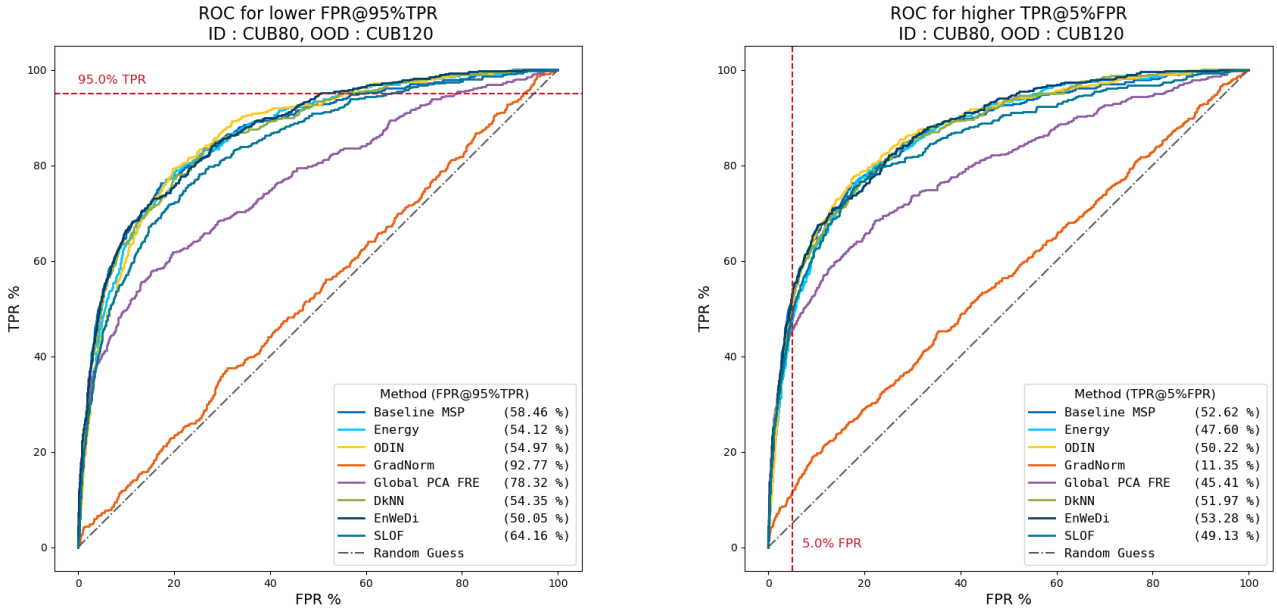
FIGURE 5.4: ROC curves for achieving the lowest FPR@TPR95 (left) and highest TPR@FPR5 (right) for each method for the task for near-OOD detection on the Papilionoidae50 dataset

Although the DkNN method performed worse than EnWeDi and Global PCA FRE methods in terms of AUROC when set with the right hyperparameters, it achieves the highest TPR@FPR5 (75.94%). This means that the DkNN method correctly identifies 75 out of 100 ID images as ID, while wrongly identifying only 5 out of 100 OOD images as ID, resulting in a false alarm rate of 5%. Another important observation from the ROC curves in Figure 5.4b is that the DkNN method would achieve a high true positive rate ($\sim$ 35%) even at a stricter false positive rate of 1%. After the DkNN method, the EnWeDi and Global PCA FRE methods achieve a high TPR@FPR5 of 63.10 and 60.70% respectively.

Out of 100 ID images, the GradNorm method can classify 10 more ID images correctly than a random guess classifier.

5.1.2.2 CUB80 Dataset

Figures 5.5a and 5.5b show the ROC curves for achieving the best possible FPR@TPR95 and TPR@FPR5 by each method for the task of near-OOD detection on the CUB80 dataset respectively.



(A) FPR@TPR95 (lower is better) (B) TPR@FPR5 (higher is better)

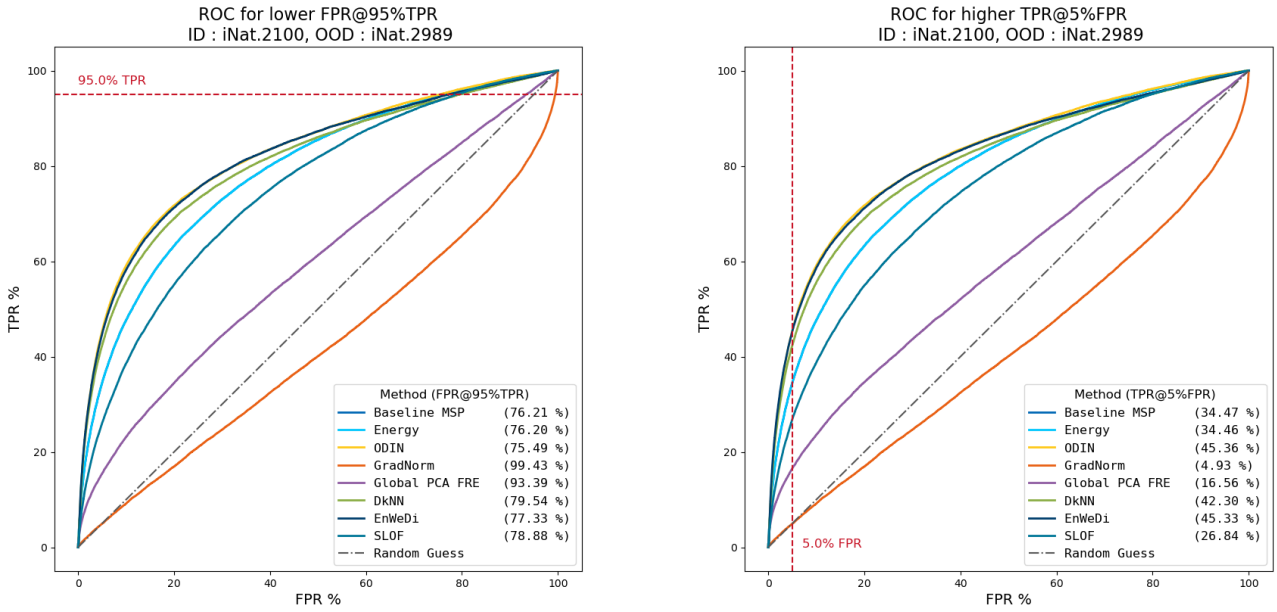
FIGURE 5.5: ROC curves for achieving the lowest FPR@TPR95 (left) and highest TPR@FPR5 (right) for each method for the task for near-OOD detection on the CUB80 dataset

The EnWeDi method again outperforms all the methods in terms of achieving the lowest FPR@TPR95 (50.05%). ODIN, Energy and DkNN perform similarly with an FPR@TPR95 of ∼54%. GradNorm performs similarly to a random guess classifier and its ROC curve conforms with the AUROC of 55.77% and FPR@TPR95 of 92.77%. With an FPR@TPR95 of 78.32%, the Global PCA FRE method is only better than GradNorm.

In terms of TPR@FPR5, there is not a significant difference between the methods for the task of near-OOD detection on the CUB80 dataset. The EnWeDi achieves the highest TPR@FPR5 (53.28%). Surprisingly, the Baseline MSP method performs outperforms all the remaining methods with a TPR@FPR5 of 52.62%. Although most of the methods, except GradNorm and Global PCA FRE, achieve a TPR@FPR95 of around 47%-52%. This difference may not be significant. As an example, the DkNN method can identify one more ID image as ID for every 100 ID images than the ODIN method at a threshold that achieves 5% FPR in those methods.

5.1.2.3 iNaturalist2100 Dataset

Figures 5.6a and 5.6b show the ROC curves for achieving the best possible FPR@TPR95 and TPR@FPR5 by each method for the task of near-OOD detection on the iNaturalist2100 dataset.



(A) FPR@TPR95 (lower is better) (B) TPR@FPR5 (higher is better)

FIGURE 5.6: ROC curves for achieving the lowest FPR@TPR95 (left) and highest TPR@FPR5 (right) for each method for the task for near-OOD detection on the iNaturalist2100 dataset

The ODIN method achieves the best (lowest) FPR@TPR95 of 75.49 % on the iNaturalist2100 dataset. The Baseline MSP and the Energy methods achieve similar but slightly worse FPR@TPR95. The ODIN method performs similarly to the EnWeDi method at most of the operating points but achieves a 3% lower FPR@TPR95. In terms of the TPR@FPR5 metric, ODIN outperforms the EnWeDi method by a marginal (∼0.5%) value. The Energy and the Baseline MSP methods are outperformed by the ODIN, EnWeDi and DkNN methods. Overall, for the iNaturalist2100 dataset, considering AUROC, FPR@TPR95 and TPR@FPR5, the ODIN method performs the best but only marginally better than the EnWeDi method.

5.2 Anomaly Detection

For anomaly detection, the out-of-distribution (OOD) dataset is the set of in-distribution (ID) test images with various corruptions applied to make the ID images 'anomalous'.

5.2.1 Long-Tailed Dataset

The results for the anomaly detection on the Papilionidae50 dataset are shown in Table 5.1.

Anomaly detection on the Pap.50 dataset (TPR@FPR1)								
Methods	**Iso noise** (ψ : 0.70)	**Motion blur** (ψ : 0.68)	**Defocus** (ψ : 0.69)	**Coarse dropout** (ψ : 0.69)	**Fog** (ψ : 0.71)	**Rain** (ψ : 0.68)	**Snow** (ψ : 0.78)	**Average** (ψ : 0.70)
Baseline MSP	72.99	0.53	0.00	18.98	5.62	63.64	35.83	28.23
Energy (T = 1)	73.26	0.00	0.00	27.81	0.00	64.71	0.53	23.76
ODIN (T=1.0, ϵ=0.001)	89.04	9.89	0.00	14.17	10.70	60.16	71.66	36.52
GradNorm (T=1)	55.08	0.00	0.00	42.78	0.00	90.64	0.00	26.93
Global PCA FRE (σ=0.99)	98.66	99.47	100.00	99.47	100.0	100.0	96.79	99.20
DkNN (k=2)	99.47	96.79	98.40	100.00	99.47	100.0	99.72	99.12
EnWeDi (k=250)	99.73	92.78	99.73	99.20	96.79	100.00	99.73	98.28
SLOF (k=23)	95.71	97.34	100.00	98.44	98.61	98.94	98.10	97.92

TABLE 5.1: Anomaly detection results on a long-tailed dataset. ID: Papilionidae50 test set, OOD: ID with corruptions added. The OOD detection performance is measured as TPR at a fixed FPR of 1%. ψ indicates the the domain similarity score between the OOD dataset and the test set of the ID dataset

For the long-tailed Papilionidae50 dataset, the output score-based methods perform poorly for the task of anomaly detection at a strict requirement of 1% FPR. For instance, at 1% FPR, GradNorm fails to classify any non-anomalous images as non-anomalous for 4 out of 7 types of corruptions. However, it can not be straightaway inferred that these methods are incapable of distinguishing between anomalous and non-anomalous images. The reasons for this are discussed in Subsection 6.2.2.

All the feature embeddings-based methods achieve a TPR@FPR1 of more than 90% across different corruptions. Specifically, the Global PCA FRE method achieves the highest average TPR@FPR1. The Global PCA FRE method and the DkNN method also exhibit a consistent performance in anomaly detection for the Papilionidae50 with TPR@FPR1 more than 96% across corruptions.

5.2.2 Balanced Dataset

The results for the anomaly detection on the CUB80 dataset are shown in Table 5.2. On the balanced CUB80 dataset, the discrepancy in anomaly detection capability between the score-based and feature embeddings-based methods is relatively smaller when compared to that on the long-tailed Papilionidae50 dataset. The Global PCA FRE method outperforms all the methods in all the cases of corruption, except one (with a very small margin of 0.2%).

Anomaly detection on the CUB80 dataset [TPR@FPR1]								
Methods	**Iso noise** (ψ : 0.77)	**Motion blur** (ψ : 0.79)	**Defocus** (ψ : 0.79)	**Coarse dropout** (ψ : 0.80)	**Fog** (ψ : 0.77)	**Rain** (ψ : 0.77)	**Snow** (ψ : 0.81)	**Average** (ψ : 0.79)
Baseline MSP	23.80	10.48	36.90	30.57	34.28	31.44	47.82	30.76
Energy (T = 10)	23.80	6.77	37.77	35.15	31.22	28.60	55.24	31.22
ODIN (T=1.0, ϵ=0.001)	24.89	7.64	16.59	26.64	23.14	18.56	52.18	24.23
GradNorm (T=10)	7.86	7.42	10.26	10.26	19.21	19.00	9.39	11.91
Global PCA FRE (σ=0.9)	48.47	23.58	62.45	56.33	67.90	67.25	60.26	55.18
DkNN (k=4)	33.41	19.21	39.74	43.89	55.02	54.15	55.02	42.92
EnWeDi (k=47)	36.90	23.14	46.51	37.77	58.30	56.55	58.30	45.35
SLOF (k=19)	35.15	16.16	46.07	41.05	50.87	57.42	60.48	43.89

TABLE 5.2: Anomaly detection results on a balanced dataset (CUB80 dataset). ID: CUB80 test set, OOD: ID with corruptions added. The OOD detection performance is measured as TPR at a fixed FPR of 1%. ψ indicates the the domain similarity score between the OOD dataset and the test set of the ID dataset

Another interesting observation is that in a set of ID images with and without motion blur, at 5% FPR, the methods could identify less than 25% of ID images as belonging to ID.

5.3 Intermediate and Far Out-of-Distribution Detection

For the intermediate- and far-OOD detection, almost all the methods show a great capability of detecting intermediate- and far-OOD images across the three ID datasets. Hence, comparing them on the AUROC metric did not provide much information to draw any conclusion. As mentioned earlier (in Section 4.6), the methods are compared at TPR@1%FPR metric in this section.

5.3.1 Papilionidae50 Dataset

For the Papilionidae50 dataset, the Costa Rica moths dataset is used as an intermediate-OOD dataset. The Places365, DTD (textures), Gaussian noise and ChestXRay datasets are considered to be far-OOD datasets where increasing levels of OODness are in that order. The results are shown in Table 5.3.

Intermediate and Far OOD on Pap.50 dataset (TPR@1%FPR)						
Methods	**Costa Rica Moths** (ψ : 0.71)	**Places365** (ψ : 0.64)	**DTD** (ψ : 0.63)	**Gaussian noise** (ψ : 0.59)	**ChestXRay** (ψ : 0.60)	**Average** (ψ : 0.63)
Baseline MSP	13.10	22.99	51.60	99.73	61.76	49.84
Energy (T = 1)	0.00	21.66	64.71	100.00	76.20	52.51
ODIN (T=1.0, ϵ=0.001)	77.27	49.73	82.35	100.00	77.81	77.43
GradNorm (T=1)	0.00	1.60	93.32	100.00	97.06	58.40
Global PCA FRE (σ=0.99)	99.73	100.00	100.00	100.00	100.00	99.95
DkNN (k=2)	100.00	100.00	100.00	100.00	100.00	100.00
EnWeDi (k=250)	99.73	99.73	100.00	100.00	100.00	99.89
SLOF (k=23)	98.13	99.73	99.73	100.00	100.00	99.52

TABLE 5.3: Intermediate- and far-OOD results for the Papilionidae50 dataset measured using the TPR@FPR1 metric. The ψ under each dataset name indicates the domain similarity score between the OOD dataset and the test set of the ID dataset.

For the intermediate-OOD case, a sharp contrastive range of TPR@FPR1 values can be seen among the methods. Four out of 8 methods achieve 100% TPR@FPR1, while 2 (Energy and GradNorm) out of 8 methods have a 0% TPR@FPR1 value. The remaining methods exhibit well-spread-out TPR@FPR1 values. The ROC curves, shown in Figure 5.7, reveal more information. It can be seen that although the three least performing methods - Baseline MSP, Energy and GradNorm have AUROC values in the range of 70%-80%, their ROC curves follow different paths to achieve this AUROC. For GradNorm, using any decision thresholds to achieve an FPR of less than 18% will lead to a 0% TPR. For Energy, this cut-off is at around 8% of FPR. Furthermore, from the ROC curves, it can be seen that the GradNorm method can achieve a lower FPR at a TPR value of 95% when compared to that of the Energy method.

For the far-OOD case, the feature-based methods outperform the score-based methods at all levels of OODness. An interesting finding is that a lower TPR@FPR1 value is achieved by most of the methods for the Gaussian noise images than for the chest X-ray images. The ChestXRay dataset is slightly more similar (higher ψ) to the Papilionidae50 dataset than the Gaussian noise dataset.

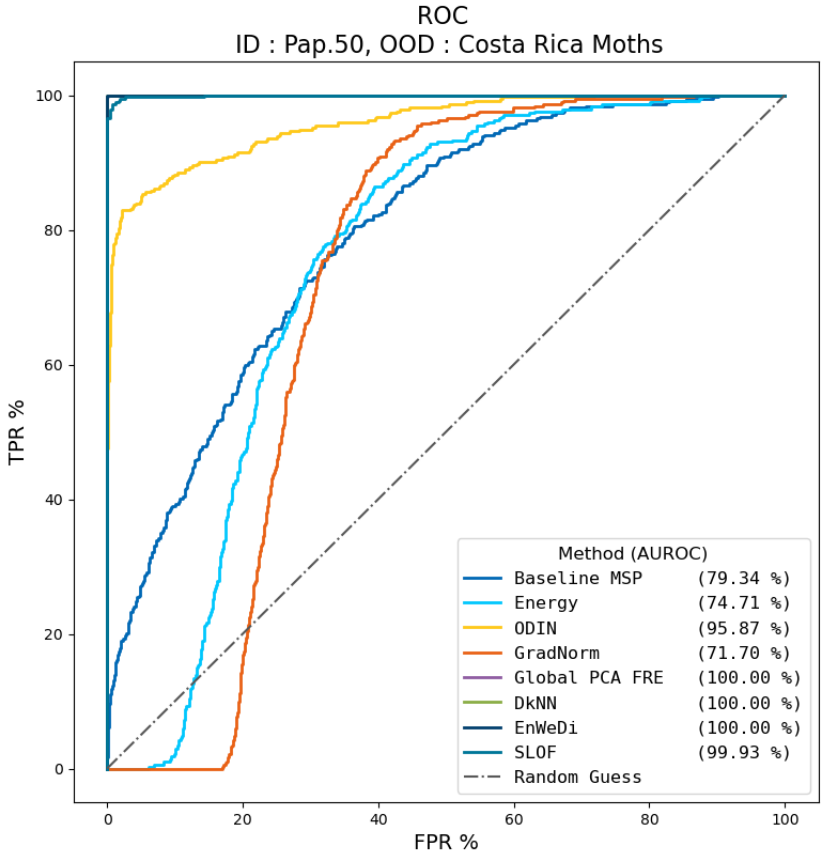


FIGURE 5.7: ROC curves for the case of intermediate-OOD detection on the Papilionidae50 dataset.

5.3.2 CUB80 Dataset

For the CUB80 dataset, the Places365 dataset is considered as belonging to intermediate-OOD. The DTD (textures), Gaussian noise and ChestXRay, in that order, are considered to be far-OOD. The results are shown in Table 5.4.

Even at a low FPR of 1%, all the methods except the Baseline MSP method achieve a TPR value of above 90% across different levels of 'OODness'. Except for the density-based method (SLOF), the Gaussian noise proved to be very easily distinguishable from OOD by all the methods.

Intermediate- and Far-OOD on CUB80 dataset (TPR@1%FPR)					
Methods	**Places365** (ψ : 0.75)	**DTD** (ψ : 0.73)	**Gaussian noise** (ψ : 0.70)	**ChestXRay** (ψ : 0.68)	**Average** (ψ : 0.71)
Baseline MSP	90.83	87.99	100.00	98.91	94.43
Energy (T = 10)	99.13	98.25	100.00	100.00	99.35
ODIN (T=1.0, ϵ=0.001)	98.25	97.38	100.00	99.56	98.80
GradNorm (T=10)	93.67	94.54	100.00	100.00	97.05
Global PCA FRE (σ=0.9)	99.78	99.78	100.00	100.00	99.89
DkNN (k=4)	99.78	99.78	100.00	100.00	99.89
EnWeDi (k=47)	99.56	98.25	100.00	99.34	99.29
SLOF (k=19)	91.70	91.05	96.29	92.58	92.91

TABLE 5.4: Intermediate- and far-OOD results for CUB80 dataset measured using the TPR@FPR1 metric. The ψ under each dataset name indicates the domain similarity score between the OOD dataset and the test set of the ID dataset.

5.3.3 iNaturalist2100 Dataset

For the iNaturalist2100 dataset, the Places365 dataset and the DTD (textures) dataset are considered intermediate-OOD. The Gaussian noise and ChestXRay datasets are considered far-OOD datasets.

Table 5.5 shows the TPR@FPR1 achieved by the OOD detection methods for intermediate- and far-OOD cases on the iNatrualist2100 dataset. For the intermediate-OOD case, unlike the Papilionidae50 and CUB80 datasets, on the iNaturalist2100 dataset, none of the methods achieve a TPR higher than 70% at a 1%FPR.

The DkNN and EnWeDi methods perform similarly to each other and achieve the highest TPRs at a 1% FPR. The ODIN method, which outperformed both of these methods for the near-OOD case, performs significantly worse in this case. Interestingly, the Baseline MSP method outperforms the ODIN method on 3 out of 4 OOD datasets.

While the Global PCA FRE method performed poorly (with an AUROC of 59.44%) on the near-OOD detection task (Figure 5.3 in Section 5.1), it performs competently on the far-OOD detection task and even outperforms all the methods for Gaussian noise dataset.

An interesting pattern seen among all three datasets is that a significant number of methods showed a lower TPR when tested on chest X-ray images as OOD than on Gaussian noise.

Intermediate and Far OOD on iNat.2100 dataset(TPR@1%FPR)					
Methods	**Places365** (ψ : 0.76)	**DTD** (ψ : 0.74)	**Gaussian noise** (ψ : 0.69)	**ChestXRay** (ψ : 0.68)	**Average** (ψ : 0.72)
Baseline MSP	51.74	55.40	71.29	66.43	61.22
Energy (T = 100)	8.17	2.36	47.69	15.35	18.39
ODIN (T=1.0, ϵ=0.001)	37.01	46.76	83.22	40.00	51.75
GradNorm (T=10)	5.82	1.77	40.90	11.41	14.97
Global PCA FRE (σ=0.97)	38.48	60.83	99.99	83.10	70.60
DkNN (k=1)	69.30	76.00	89.37	97.28	82.99
EnWeDi (k=15)	69.29	76.01	89.37	97.29	82.99
SLOF (k=4)	8.39	17.08	12.96	47.49	21.48

TABLE 5.5: Intermediate- and far-OOD results for the iNaturalist2100 dataset were measured using the TPR@FPR1 metric. The ψ under each dataset name indicates the domain similarity score between the OOD dataset and the test set of the ID dataset.

5.4 Effect of the Amount of In-Distribution Data Used on Feature Embeddings-based Methods

In this section, the results for feature-based OOD detection methods with a limited set of reference ID data are presented.

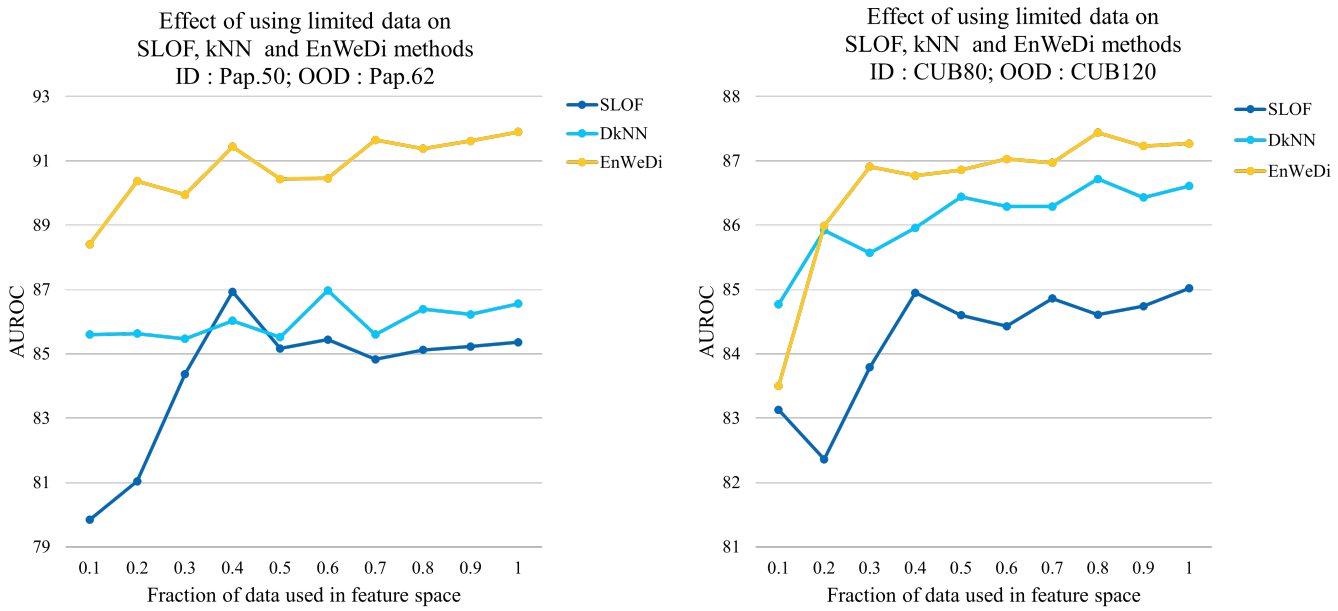
5.4.1 Distance-, Entropy- and Density-based Methods

Figure 5.8 shows the OOD detection performance of the distance-based (kNN), density-based (SLOF), and entropy-weighted distance-based (EnWeDi) methods with limited data in the feature space (kNN search space). Figures 5.8a and 5.8b show the cases of long-tailed (Papilionidae50) and balanced (CUB80) in-distribution datasets respectively.

The density-based method (SLOF) shows significant improvement with adding more ID reference data until 40% of the data is added. For the distance-based DkNN method, this highly depends on the ID dataset. In the case of a long-tailed ID dataset, the DkNN shows only a marginal improvement in OOD detection when adding more data into the kNN search space. On the balanced dataset, doing the same shows a noticeable increase in the OOD detection performance. However, the gain diminishes as more data is added. For the EnWeDi method, which is distance-based and entropy-based, having more data points in feature space leads to an increase in OOD detection accuracy on long-tailed as well as balanced datasets.

Overall, it can be seen that adding ID reference data to the feature space increases the near-OOD detection performance for the distance-, entropy- and density-based methods.

However, the methods exhibit diminishing returns in terms of OOD detection capability after a certain amount of data in the feature space.



(A) case: near-OOD on long-tailed ID dataset

(B) case: near-OOD on balanced ID dataset

FIGURE 5.8: Effect of using limited ID reference data on the distance, density, and entropy-weighted-distance based methods. A 0.1 on the x-axis implies that 10% of the training data is used in feature space

5.4.2 Feature Reconstruction Error-based Methods

The results of varying the amount of data used for modelling the subspace of the in-distribution show interesting results depending on the method (per class or global) and the type of dataset it is tested on. The results for near-OOD detection for Global and Per-class PCA FRE methods are shown in Figures 5.9 and 5.10 respectively. The x-axis on the plots is the variance (σ) that needs to be maintained in a feature embedding when reducing it to fewer dimensions using PCA.

Global PCA FRE Method: For the Global PCA FRE method, increasing the amount of data for modelling the ID subspace results in an increase in AUROC on the Papilionidae50 dataset. As can be seen in Figure 5.9, the increase depends on the value of σ chosen. It is also seen that after a certain value of σ, the OOD detection (in AUROC) drops. The higher the amount of data used, the lower the value of σ after which the AUROC drops. However, for the balanced CUB80 dataset, this does not apply. The OOD detection performance does not improve much ($<$4%) by using more data to model the ID subspace. Surprisingly for the CUB80 dataset, using 10% of data for training the PCA model gives similar results as using 100% data.

Per-class PCA FRE Method: For the Per-class PCA FRE method, using more data for subspace modelling shows higher detrimental effects on OOD detection on the long-tailed Papilionidae50 dataset at higher σ values. This implies that for the Per-class PCA

FRE method, choosing a lower value for the σ hyperparameter results in better OOD detection. As shown in Figure 5.10, the Per-class PCA FRE method, when tested on the CUB80 dataset using more data to model the ID subspace, does not show much improvement irrespective of the value of the hyperparameter (σ) chosen. Increasing the amount of data for subspace modelling results in less than 3% improvement in AUROC on the CUB80 dataset.

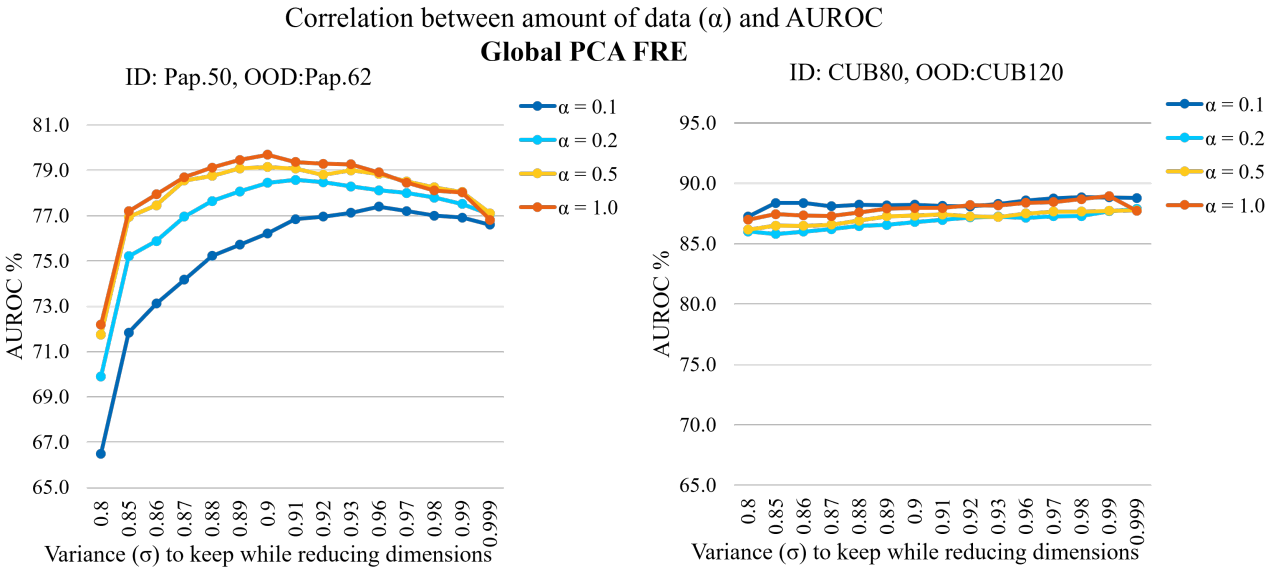


FIGURE 5.9: Effect of using limited data to model the in-distribution subspace for the Global PCA FRE method. An α of 0.1 implies 10% of the ID data is used for subspace modelling.

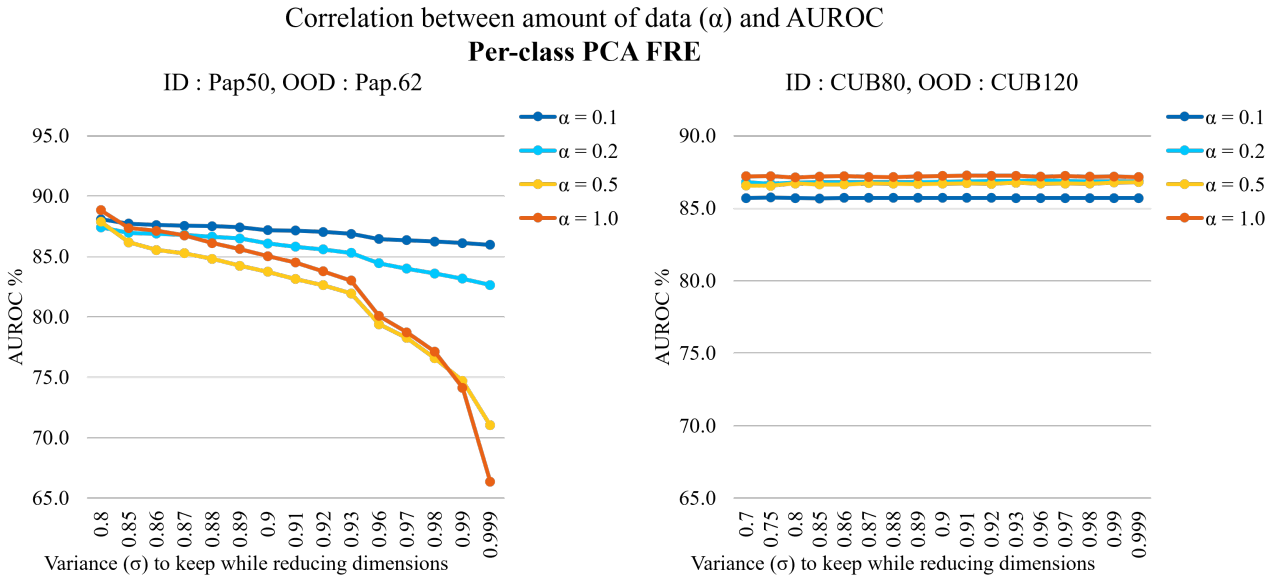


FIGURE 5.10: Effect of using limited data to model the in-distribution subspace for Per-class PCA FRE method. An α of 0.1 implies 10% of the ID data is used for subspace modelling.

In conclusion, for very small datasets (around 400 images), the PCA FRE-based methods can be used for OOD detection safely only if the datasets are balanced. The results for a related topic (Per-class versus Global PCA) are presented in Section 5.6.

5.5 Inference Time of OOD Detection Methods

Figure 5.11 shows the per-image OOD detection inference time with respect to the AUROC achieved by the methods on the iNaturalist2100 dataset.

5.5.1 Score-based methods

The Baseline MSP method and Energy method are the fastest out of the selected methods. The OOD detection inference time of these methods is approximately equal to the time taken by a forward pass on the CNN model used. The ODIN and GradNorm methods, which require the calculation of gradients with respect to the output layer of the CNN model, take significantly more time to calculate the OOD detection score when compared to other score-based methods. The inference times for ODIN and GradNorm are 160.2 ms and 109.1 ms respectively. It was observed that the score-based methods exhibit similar inference times across the ID datasets and hyperparameters.

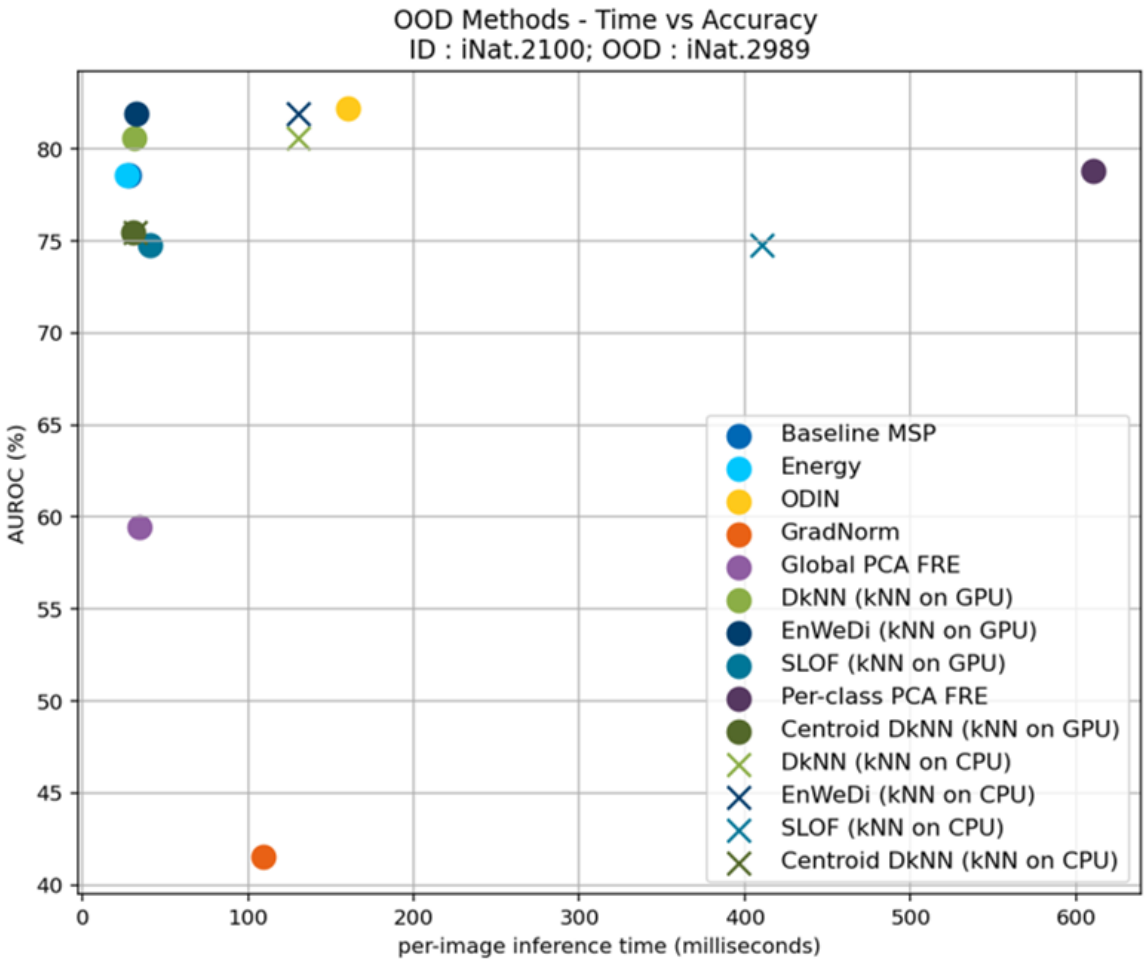


FIGURE 5.11: Inference times of OOD detection methods (and their variants) on the iNaturalist2100 dataset

5.5.2 kNN-based methods

On the iNaturalist2100 dataset, when the kNN search for the DkNN and EnWeDi method is performed on a GPU, the inference time (per image) is approximately 4 milliseconds more than a forward pass on the CNN model. When the kNN search for these methods is performed on a CPU instead of a GPU, the OOD detection is around 4 times slower on the iNaturalist2100 dataset. For instance, the inference time is 130.0 ms instead of 31.4 ms for the DkNN method.

However, the significant increase in the inference time when kNN search is performed on the CPU instead of a GPU is only observed when the ID dataset is large such as the iNaturalist2100 dataset (for which there are around 0.2 million feature embeddings in the kNN search space). For instance, on the Papilionidae50 dataset, when the kNN search is performed using the CPU, the DkNN and the EnWeDi methods take 37.8 ms and 37.9 ms respectively, while the same using a GPU is 29.7 ms and 29.9 ms respectively.

As the DkNN method (on GPU) is already fast, the advantage of using the Centroid-DkNN method in terms of OOD detection speed is marginal. On the iNaturalist2100 dataset, the Centroid-DkNN method takes 30.6 ms for OOD detection per image (which is 0.8 ms less than the DkNN method for the same). Similarly, using less amount of ID data in feature space also does not lead to any significant improvement in inference speed. Interestingly, the Centroid-DkNN method takes only 31.7 ms (around 1 ms more) when the kNN search is performed on a CPU instead of a GPU. The density-based method, SLOF, is slower than the distance-based methods (DkNN, Centroid-DkNN and EnWeDi). However, the difference is significant only when the kNN search is performed on a CPU for a large dataset such as the iNaturalist2100 dataset.

5.5.3 PCA-based methods

With an average inference time of 34.5 ms, the Global PCA FRE method is slightly slower (by 3 ms) than the DkNN and EnWeDi methods. The Global PCA FRE method's inference time is similar across the three ID datasets. However, its per-class variant, the Per-class PCA FRE method's inference time largely depends on the ID dataset. More specifically, the inference time is directly proportional to the number of classes in the dataset. While this method takes 610.0 ms on the iNaturalist2100 dataset, it takes 51.2 ms on the CUB80 dataset.

5.6 Per-Class and Global Approach to Distance-based and Feature Reconstruction Error-based Methods

The distance-based method and the PCA FRE-based methods can also be implemented at a class level instead of at the dataset (global) level. The Per-class PCA FRE method is described in Subsection 3.2.5.2 and the DkNN method at a class level is the Centroid-DkNN method (described in Subsection 3.2.6.3). The comparison of applying the two approaches for the task of near-OOD detection is shown in Figure 5.12.

For the long-tailed Papilionidae50 data set, it can be inferred that applying the DkNN and PCA FRE approaches at the dataset level instead of the class level provided better OOD accuracy. However, for the balanced CUB80 dataset, applying these methods at the class level (instead of the dataset level) leads to an improvement in OOD accuracy. For the kNN-based method, the improvement is insignificant. The improvement is more noticeable for the PCA FRE-based approach where the increase in AUROC from the global to per-class method is more than 7%. The reason why applying these approaches per class for

OOD detection is beneficial for one type of dataset and detrimental for another is discussed in Section 6.6.

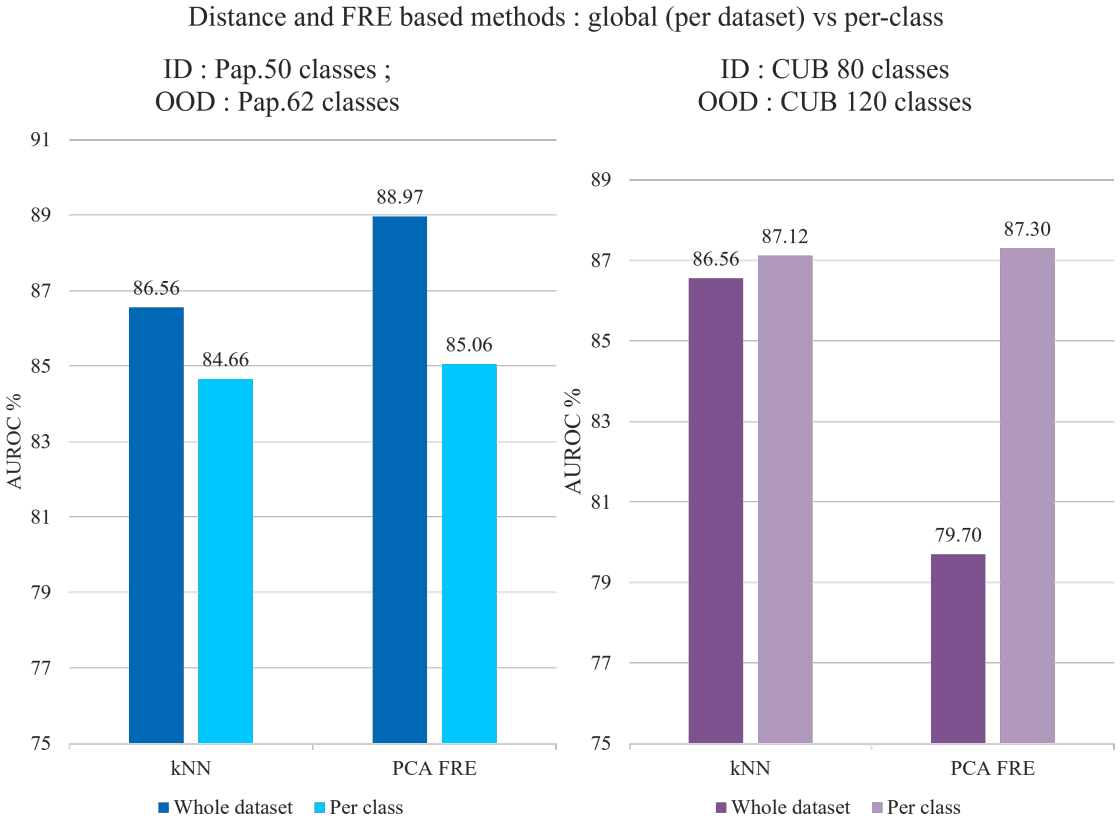


FIGURE 5.12: Effect of applying feature embeddings-based methods - DkNN and feature reconstruction error based methods globally (entire ID dataset as a single entity) and per-class. The results are shown for the task of near-OOD detection on the Papilionidae50 (left) and CUB80 (right) datasets

5.7 Effect of Stacking

As mentioned in Section 4.4, the feature embeddings can be extracted from all the 9 major layers (*layer.8* to *layer.0*) of EfficientNetV2-M [70] and stacked to form a single feature embedding. The near-OOD detection performance with penultimate layer feature embeddings and stacked feature embeddings are compared in Figure 5.13.

For the Papilionidae50 dataset, it can be observed that using stacked features improves OOD detection significantly. The range of increase in AUROC ranges from 2.2% - 6.8%, with the DkNN method getting benefited the most. However, for the CUB80 dataset, stacking the features drastically reduces the OOD detection capability of the methods. The AUROC values reduce in the range of 6.4% to 12.3%. Interestingly, with stacking, the DkNN method which benefited the most on the Papilionidae50 dataset, gets significantly affected by the same on the CUB80 dataset - a huge drop of 12.3% in AUROC. This discrepancy in OOD performance between datasets is discussed and a method to avoid the degradation of OOD performance when stacked is proposed in Section 6.5.

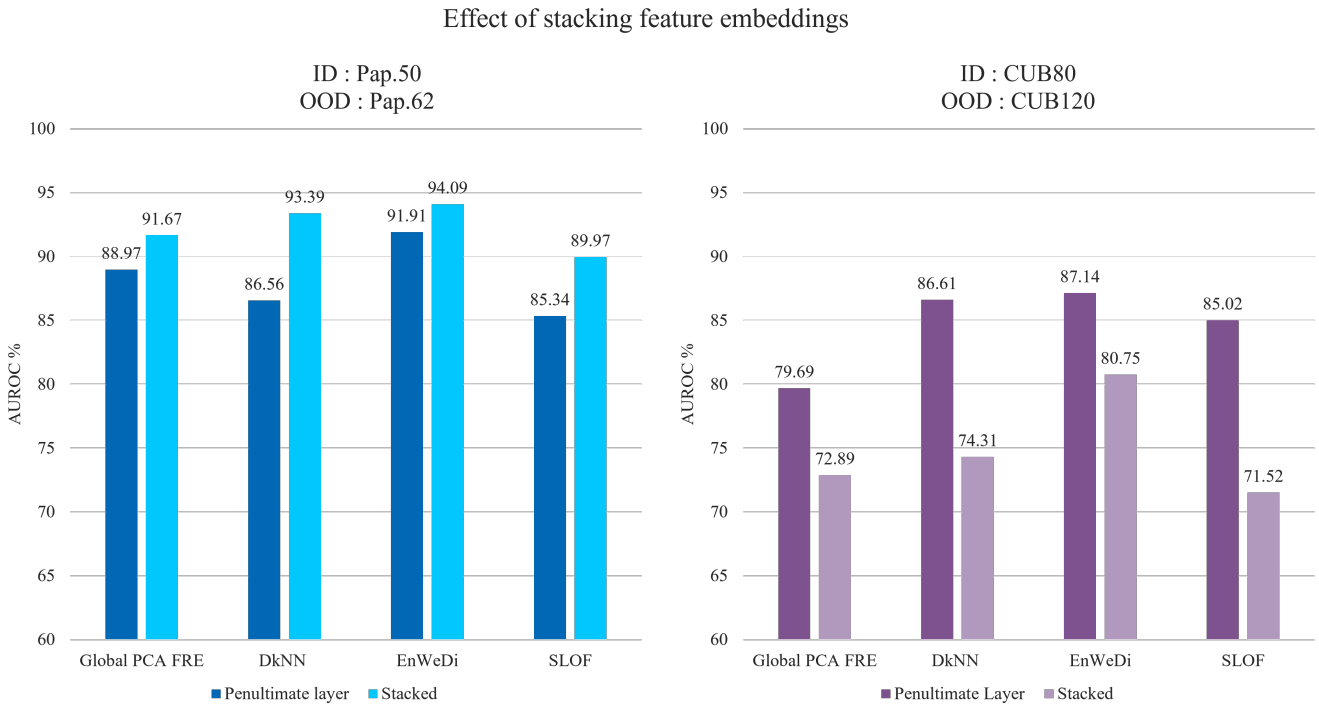


FIGURE 5.13: Effect of stacking the feature embeddings on near-OOD detection on the Papilionidae50 (left) and CUB80 (right) datasets.

Chapter 6

Discussion

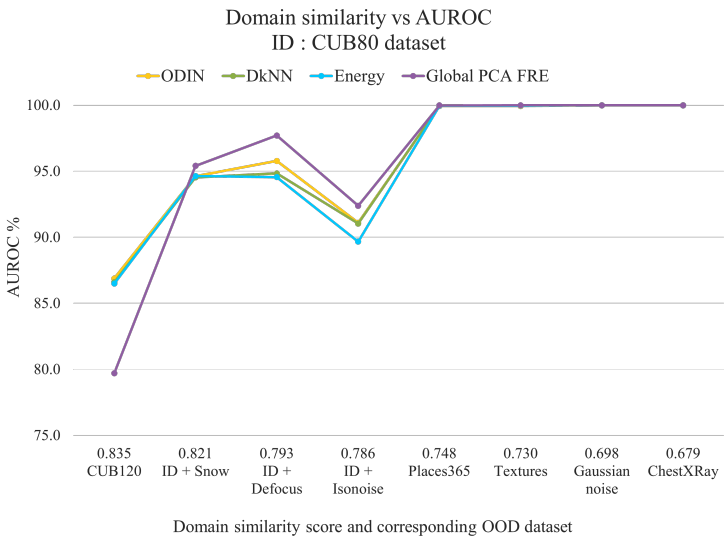
In this chapter, most of the comparisons and discussions around the results are focused on the near-OOD detection task as this is a more challenging task. Moreover, a classifier deployed in an open world may encounter images from near-OOD more often than images from far-OOD. For instance, a system that is trained to classify preserved butterfly species in a biodiversity centre is expected to encounter new species of butterfly species more often than species of completely different insects. Furthermore, as shown in Section 5.3, many methods can achieve a high AUROC on the intermediate- and far-OOD detection tasks but, but do not perform as well for near-OOD detection tasks.

6.1 Suitability of Using Domain Similarity for Measuring OOD-ness

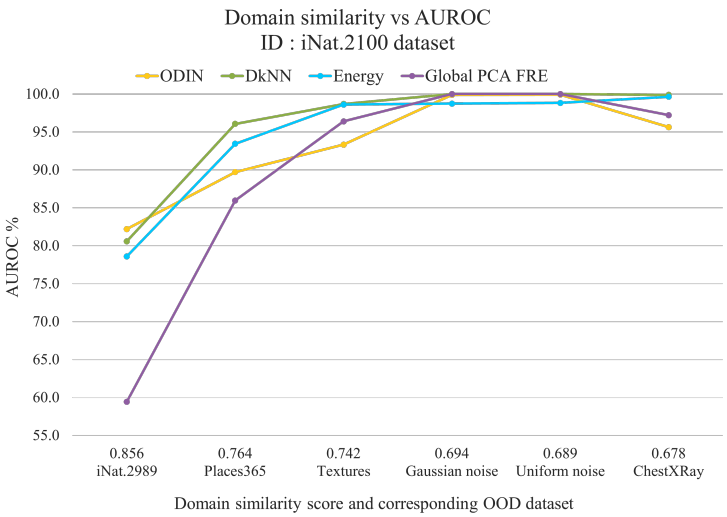
This section discusses the approach used for answering the research question: *"For a given in-distribution dataset, how can the difficulty of detecting OOD images from a particular OOD dataset be quantified?"*

To quantify the OOD detection difficulty, the concept of domain similarity between in-distribution and out-of-distribution was used. For a given ID dataset, the domain similarity measures to a set of OOD datasets were in coherence with the subjective categorising of them into near-, intermediate- and far-OOD. For instance, for the CUB80 birds dataset, intuitively the order in which the OOD datasets would be most to least similar would be CUB120 (birds), CUB80 (ID) with corruptions applied, Places365 (due to backgrounds), textures (patterns) and then Gaussian noise (featureless). The domain similarity measures indicate the same order as well.

While it may not be straightforward to measure the accuracy of the domain similarity in a standardised manner, a simple pseudo-validation on the suitability of domain similarity as a measure for OOD detection difficulty was done. Figure 6.1 shows the correlation between the domain similarity between a given ID and OOD dataset and the accuracy (AUROC) of the OOD detection task between them. Two output-score based and two feature embeddings-based OOD detection methods, all of which use different concepts for OOD detection, are shown.



(A) For CUB80 ID dataset



(B) For iNaturalist2100 ID dataset

FIGURE 6.1: Domain similarity scores vs AUROC for two datasets. DkNN and Global PCA FRE are feature embeddings-based methods while Energy and ODIN are output score-based methods

The domain similarity vs AUROC curves mostly shows the expected relation - as the domain similarity between ID and OOD dataset decreases, easier it is to distinguish between the images from them and therefore higher the AUROC. As it can be seen, the domain similarity - AUROC relation mostly follows the same trend across the four methods irrespective of whether they are feature-embeddings-based, or output-score based.

6.1.1 Domain Similarity and Model Architecture Dependency

In this research, the domain similarities are calculated using feature embeddings of ID and OOD images extracted from an EfficientNetV2-M model pre-trained on ImageNet-21K [8].

To check if the domain similarity is dependent on the model architecture, it was also calculated using other models of various architectures - MobileNet V3-L [99], BiT-ResNet101V2 [100] and EfficientNetV2-L [70]. On different models, although the domain similarity scores have a different numerical range, the OOD datasets will be arranged in the same order of OODness with these domain similarity scores as well.

For instance, Figure 6.2, shows the domain similarity scores calculated for the Papilionidae50 dataset and corresponding OOD datasets. Except at one instance (MobileNet-v3-L between ChestXRay dataset and Gaussian noise dataset), all the OOD datasets follow a similar trend of domain similarity across different architectures. The domain similarity scores calculated using different architectures for CUB80 and iNaturalist2100 are shown in Figures A.1 and A.2 in the appendix.

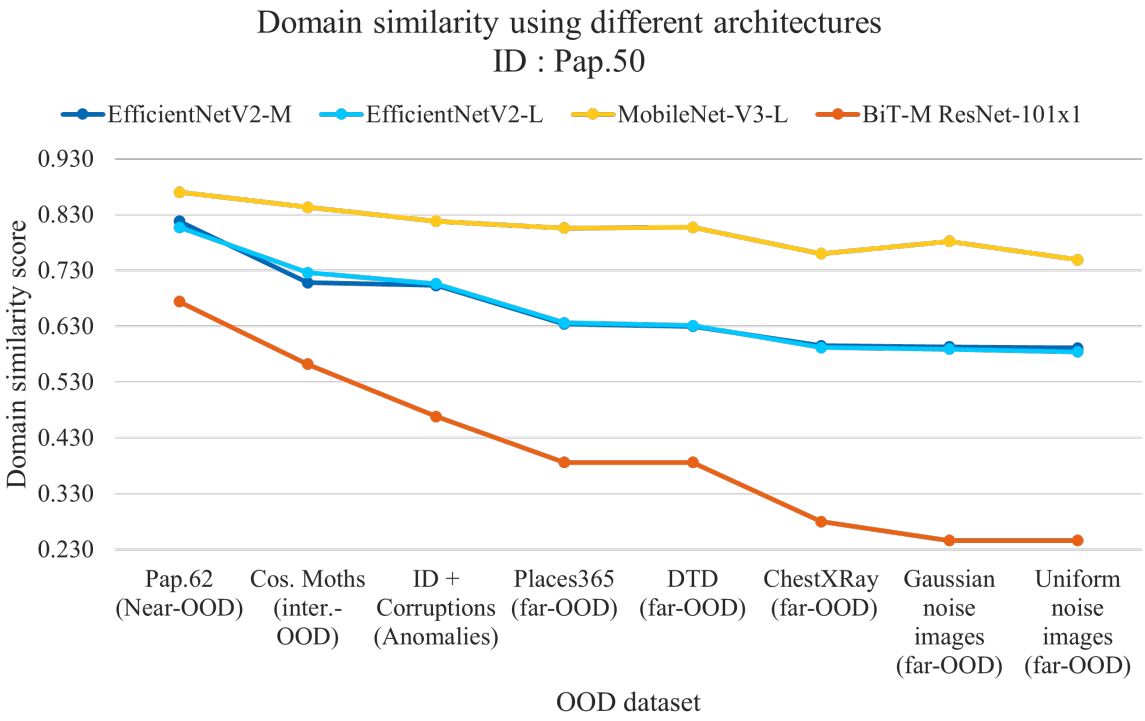


FIGURE 6.2: Domain similarity scores using different model architectures

6.1.2 Comparison With Existing Works for Measuring the Difficulty of OOD Detection

In [23], the authors propose the openness factor as an estimate for OOD detection difficulty. This factor is solely determined by the number of classes in ID and OOD datasets and not based on the similarity between them. For an ID dataset with 50 classes of bird species, the openness factor with an OOD dataset of 50 classes of new (OOD) bird species and 50 classes of x-ray images would be the same. Whereas the proposed approach of using domain similarity would represent the significant difference between them. Furthermore, the openness factor is difficult to quantify for real-world applications as it is challenging to estimate the number of possible OOD classes that could be encountered by a classifier. Hence the openness factor can vary highly based on the subjective context of making a category of OOD images into a class.

In [101], the confusion log probability (CLP) metric as a measure to estimate OOD detection difficulty is introduced. The CLP needs an ensemble of classifiers to be trained on a dataset containing a union of the ID and OOD datasets. For each combination of the ID-OOD dataset, a new ensemble of classifiers needs to be trained. This could be time (and energy) consuming and may not be ideal if the ID and/or the OOD datasets are large (or many). The proposed domain similarity approach needs only a single off-the-shelf classifier that is already trained on a wide variety of images or a classifier that is trained on the ID dataset.

Another approach to estimating OOD detection difficulty is presented in [102]. In this method, OOD detection difficulty is estimated at a class level i.e., the OOD classes within an OOD dataset are grouped into categories that indicate varying levels of OOD detection difficulty. This method is based on the assumption that if images belonging to a particular OOD class have high prediction confidences on the classifier trained on the ID dataset,

then that class is more difficult to be detected as an OOD class. This approach can be extended to measure OODness (and OOD detection difficulty) at the dataset level as well. However, the work was published towards the end of this research work (February 2023), leaving limited time to implement and compare it with the domain similarity approach.

6.2 Interesting Results on OOD Detection

In this section, a few surprising/interesting results achieved by the OOD detection methods in the previous chapter are discussed.

6.2.1 DkNN Method Achieving a 75% TPR@FPR5 on the Near-OOD Detection

On the Papilionidae50, it can be seen in Figure 5.4b that the DkNN method can achieve a very high TPR of 75% while meeting a 5% FPR requirement for the near-OOD detection case. A hypothesis on how the DkNN method achieves this is presented here.

A hypothesis that can be derived from the underlying idea behind the DkNN method is *For an input image from ID, the k-nearest neighbours in the feature space majorly consist of the feature embeddings from the same class from the ID training/reference set.* From observations, it can be said that this hypothesis holds in most cases. With this preface, consider the following scenario: a feature embedding f_x from the ID test set belongs to an ID class C_{f_x} which has $|C_{f_x}|$ number of feature embeddings in the feature space. When $k >> |C_{f_x}|$, then the probability that the k^{th} nearest neighbour for f_x is from C_{f_x} is very low.

The DkNN method achieves this high TPR when the hyperparameter k is set to 150. This means, for an input image, the OOD score assigned is the distance to its 150^{th} nearest neighbour in the feature space. In the Papilionidae50 dataset, more than 90% classes have less than 150 images. The two contradicting questions that then arise are *"Shouldn't an ID test image that has less than 150 images per class in the feature space mostly likely get classified incorrectly when $k \geq 150$?"*. If the hypothesis holds most of the cases, then *"How is a high TPR possible even though the images belonging to 90% of the classes are most likely getting misclassified?"*.

In the Papilionidae50 dataset, the training set (reference set), as well as the test set, are long-tailed. Even though 45 out of 50 classes have less than 150 images, 77% of the feature embeddings in the feature space belong to the rest 10% of the classes (5 out of 50), each of which has more than 150 images per class. The rest 23% of the feature embeddings belong to 45 classes, out of which 35 classes have less than 10 images per class. The ID test set follows a similar ratio in the distribution of images per class. Hence, the following scenario could be leading to the reported 75% TPR: At $k = 150$ a precise distance threshold can be set which is just smaller than the distance to the 150^{th} nearest neighbour for 5% of OOD images. In the ID dataset, per 100 images, 23 images belong to classes having less than 150 images per class. Even if all 23 images are misclassified as OOD, out of the remaining 77 images, 75 images are classified correctly as ID.

This is also the reason why the TPR increases very slowly after reaching 77% in the ROC. Even a small decrease in the distance threshold leads to a large number of OOD images getting classified as IDs, thereby leading to a rapid increase in the false positive rate. This can be seen in Figure 5.4b.

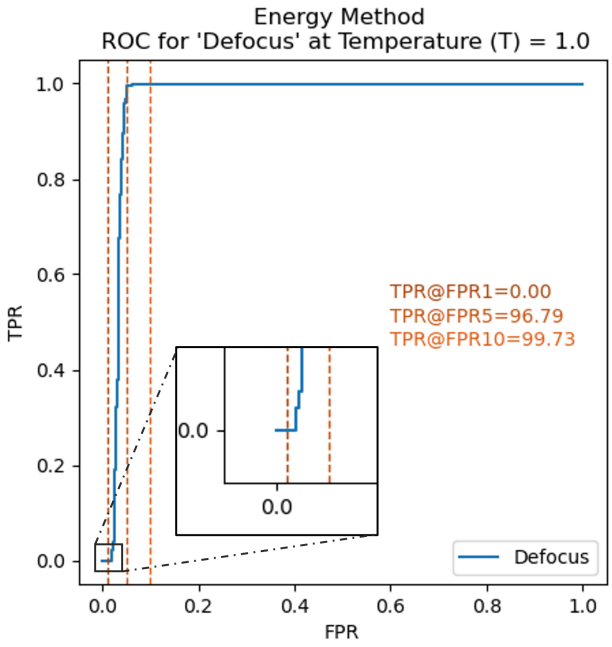
This indicates that for a long-tailed dataset, the value of k chosen for the DkNN method to achieve a true positive rate requirement, significantly depends on the dataset.

Note that the class(label) information was used only for the above discussion and the DkNN method does not require any class(label) information for OOD detection.

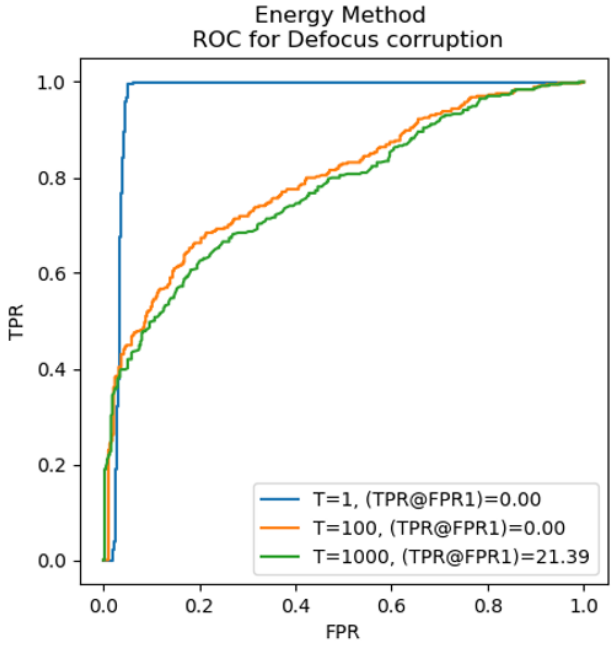
6.2.2 0% TPR@FPR1 for Anomaly Detection on the Papilionidae50 Dataset

For the task of anomaly detection on the Papilionidae50 dataset, a few methods such as GradNorm and Energy achieved 0% TPR@FPR1 as shown in Table 5.1. However, it is to be noted that methods that show a 0% TPR at 1% FPR can not be considered completely incapable of detecting anomalies for two reasons:

- Meeting the stern requirement of having an FPR $\leq$ 1% could be too harsh on some of the methods which are capable of meeting slightly relaxed FPR requirements. For instance, as shown in Figure 6.3a, the Energy method has a TPR value of 0.0 at 1% FPR, but a TPR value of 96.79% at 5% FPR.
- It is to be considered that the methods are not operated with hyperparameters that are specific to this strong requirement. It has been observed that using a different set of hyperparameters can make respective methods achieve this requirement in anomaly detection. For instance, as shown in Figure 6.3b, the energy method can achieve a non-zero TPR@FPR1 when the temperature parameter of the method is set to 1000. Although this decreases the overall anomaly-detecting capability (measured using AUROC).



(A) TPR at a slightly relaxed requirement of FPR



(B) TPR at 1%FPR at different hyperparameters

FIGURE 6.3: ROC curve for Energy method for the task of anomaly detection on ID: Papilionidae50, OOD: ID applied with Defocus corruptions.

In summary, achieving 0% TPR@FPR1 for anomaly detection is a strict requirement, and some methods may still be capable of detecting anomalies with slightly relaxed FPR

requirements or adjusted hyperparameters. Nonetheless, the trade-offs between meeting this requirement and overall anomaly-detecting capability should be considered.

6.3 Selecting the Best OOD Detection Method

In this section, using the results from the previous chapter, the research question *"What are the OOD detection methods that can consistently achieve the best accuracies over different OOD difficulties when measured across a range of evaluation metrics?"* will be discussed.

In the case of near-OOD detection, except on the CUB80 dataset, the difference in AUROC achieved by the methods is significant enough to choose one over the other. The proposed method - EnWeDi performs the best consistently across the datasets. For the iNaturalist2100, the input-perturbations-based method - ODIN outperforms the EnWeDi method (by 0.3%) for the task of near-OOD detection but does not show this performance consistently across other OOD detection tasks or datasets. Furthermore, in ODIN, for each input image, a new image with perturbations added to the input is generated, which must also be inferred from the model. This would reduce the overall throughput (number of outputs per unit time) by at least 50% of the classifier model. As shown in Figure 5.11, the ODIN method is at least 5 times slower than the EnWeDi method. If enough computation resources and memory is available, the EnWeDi method can be pipe-lined with the classifier model's inference.

When the requirement is to achieve a higher TPR at a set rate of false positives, then based on the results, both DkNN and EnWeDi methods can be chosen. However, the DkNN method achieves 10% higher TPR than the EnWeDi method on the Papilionidae50 dataset. A hypothesis on how the DkNN method achieves this interesting result was discussed in Subsection 6.2.1.

For the task of anomaly detection, the Global PCA FRE method outperforms all the methods when tested on the Papilionidae50 and the CUB80 datasets. More importantly, it achieves $\sim 10\%$ higher TPR@FPR1 than the second-best method (EnWeDi) on the CUB80 dataset. This is in coherence with the literature of the PCA-based FRE method [5], where it performs similarly to, and in a few cases outperforms the current state-of-the-art anomaly detection methods.

However, The Global PCA FRE method does not show decent OOD detection performance on a dataset having a large number of classes such as the iNaturalist2100. In the Global PCA FRE method, a single PCA model is used to model the subspace of the entire ID dataset. For datasets with a large number of categories/classes, a single subspace model is too generalised that even an image from the OOD dataset gets a low feature reconstruction error. To solve this, the Per-class PCA FRE approach can be followed, where each class's subspace is modelled individually and can lead to a better OOD performance. However, a per-class approach for the iNaturalist2100 dataset is not practical as it requires performing forward and inverse transformations of dimensionality reduction 2100 times on an input image's feature embedding. As shown in Figure 5.11, the Per-class PCA method is 18 times slower than the Global PCA FRE method in this case. The Per-class PCA FRE method also does not perform well when the ID dataset has a class with very few images such as the Papilionidae50 dataset. The reasons for this are discussed in detail in Section 6.6.

For intermediate- and far-OOD, most of the methods achieve similar and near-perfect OOD detection accuracies on the Papilionidae50 and CUB80 datasets. With iNaturalist2100 as ID and Places365 dataset as OOD, the only DkNN and EnWeDi methods achieve a TPR@FPR1 of more than 50% when 3 out of 8 methods achieve less than 5% for the

same.

Distilling the results From the results in Sections 5.1 (near-OOD detection), 5.2 (anomaly detection) and 5.3 (intermediate- and far-OOD), it is observed that not all the OOD detection tasks are equally difficult. For instance, on the CUB80 dataset, for the task for intermediate- and far-OOD detection, all the methods (except the Baseline MSP in one case), achieve true positive rates above 90% even at the strong requirement of 1% false positive rate. For such cases, the difference in OOD performance between methods is marginal and hence comparing methods is trivial. Therefore from the set of results presented in the above-mentioned sections, the two most difficult OOD detection tasks per dataset covering different types of evaluation metrics are chosen and are shown in Figure 6.4. This helps to understand the consistency of the methods across various aspects of the OOD detection tasks, across requirements and types of datasets to pick a method that performs competently overall.

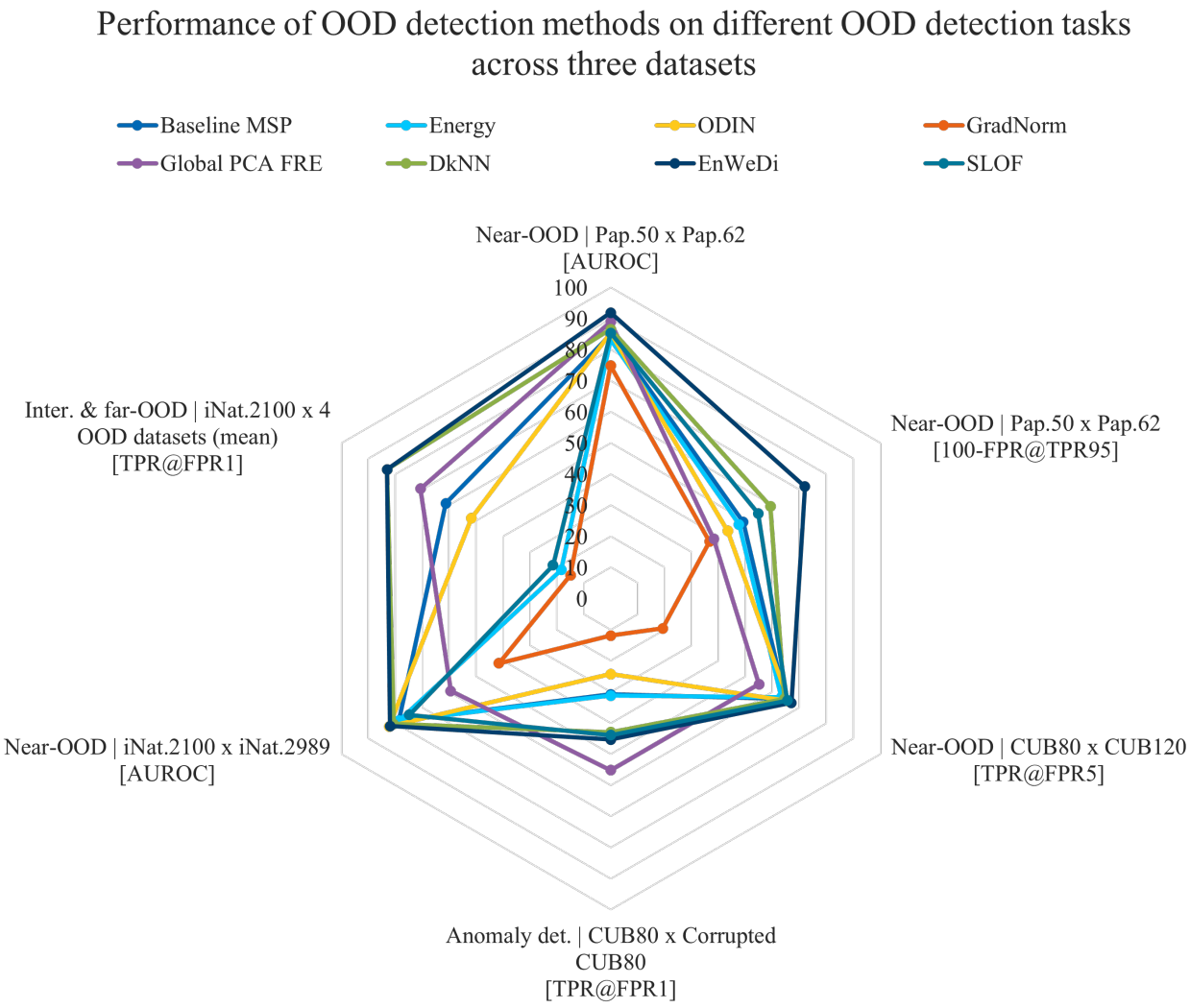


FIGURE 6.4: Radar graph showing the performance of methods across various aspects of OOD detection, types of datasets and requirements (evaluation metrics)

Overall, it can be seen that the EnWeDi method consistently performs competently across all three datasets and different evaluation metrics.

Figure 6.5 summarises the OOD detection methods recommended for different scenarios based on the results and discussion above.

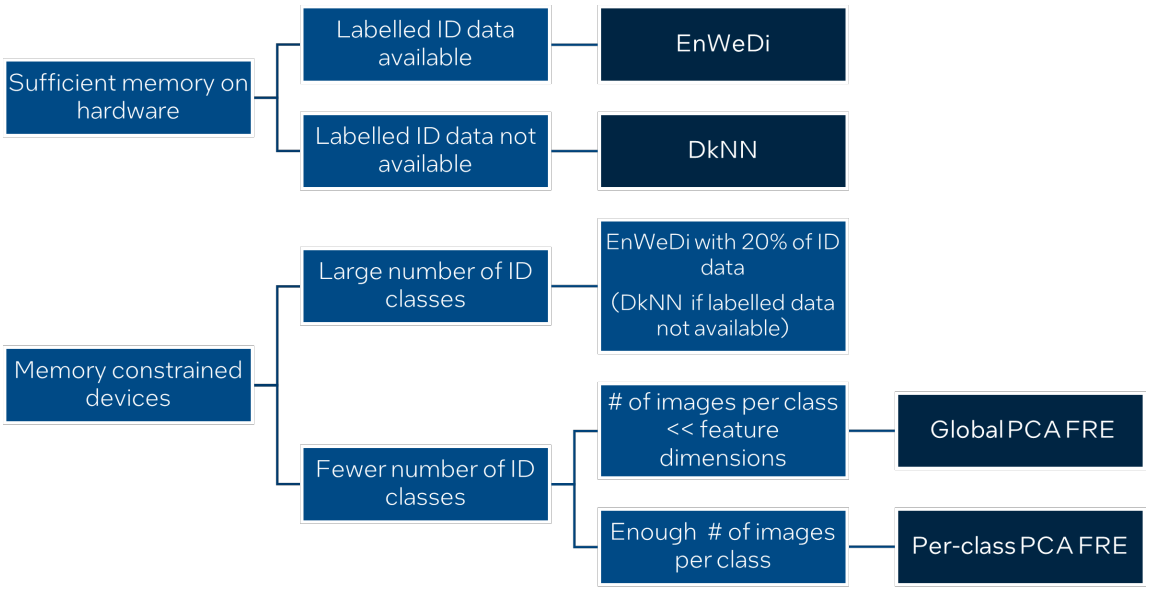


FIGURE 6.5: OOD detection method recommendations for different scenarios

Ease of implementation and retrofitting Apart from the quantitative metrics, choosing an OOD detection method based on ease of implementation, retrofitting and explainable OOD detection is discussed below. The Baseline MSP is the easiest OOD detection method as the OOD decision is made by comparing the maximum of the already calculated softmax prediction probabilities with a threshold. However, it is not the most robust OOD detection method due to the underlying assumptions it is based on. The Energy method is the next simplest method to implement as it can be retrofitted by adding the energy equation (Equation 3.2. ODIN and GradNorm require backpropagating through the last layer of the neural network to calculate gradients. All the output score-based methods do not need any ID images (expect to set the decision threshold) and hence can be added to an existing classifier even when the training (ID) data is no longer available. For all the feature embeddings-based methods, images from the ID dataset (and their labels, depending on the method) are required. For PCA FRE-based models, the ID data is not required during the testing phase, whereas the feature embeddings of the ID images are required even during the testing phase for DkNN, EnWeDi and SLOF variants. The EnWeDi method can also provide more reasoning on why a particular input image is classified as ID or OOD. This is explained in Subsection 6.4.3.

6.4 Proposed Method - Entropy Weighted Nearest Neighbours Distance

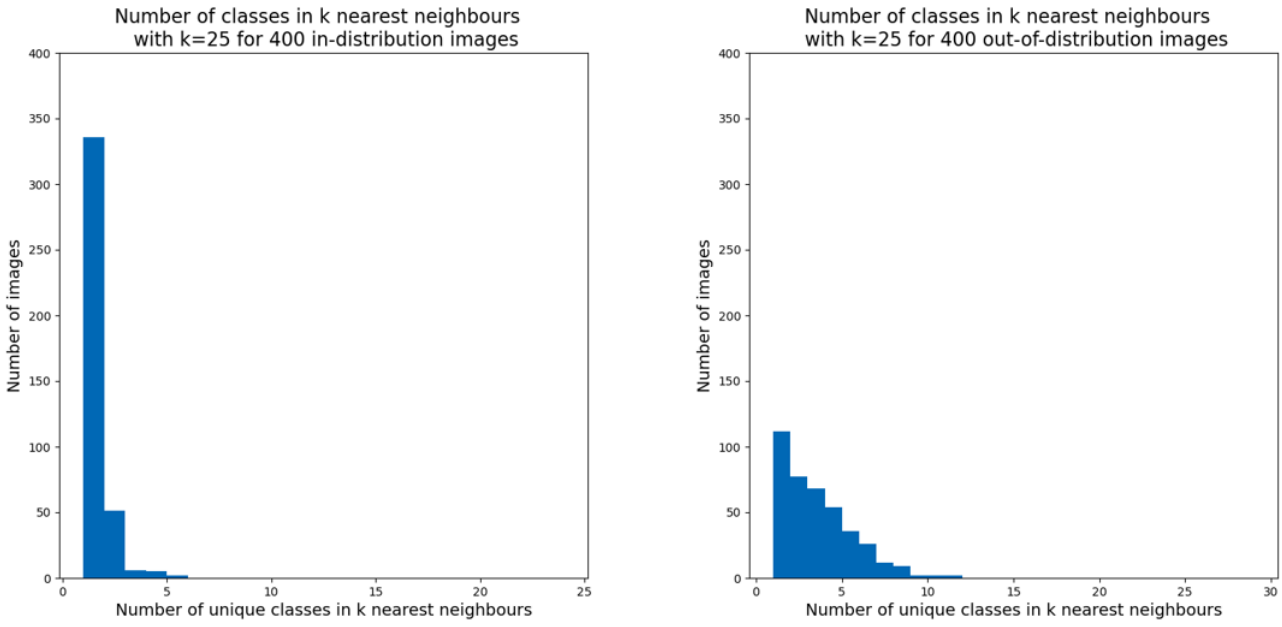
The discussion in this section is related to the research question *"What improvements can be made to the existing OOD detection methods to further improve their accuracy for the task of OOD detection on fine-grained and long-tailed in-distribution datasets?"*

In this research, the Entropy weighted nearest neighbour's distance (EnWeDi) for OOD detection is proposed. In this section, the hypothesis behind the method is validated. A few reasons with examples have been presented that show why EnWeDi is a better OOD detection method. Next, the method's ability to perform 'explainable OOD detection' is discussed. The list of limitations of the method is presented and an improvement is proposed. Finally, a few points to make the EnWeDi method more feasible to implement and use are listed.

As the DkNN method is closest conceptually and in terms of performance for most of the OOD detection tasks, the EnWeDi method is compared against it in this section.

6.4.1 Validation of the Hypothesis Behind the EnWeDi Method

The EnWeDi method is based on the hypothesis that an OOD image will have nearest neighbours belonging to multiple classes while the ID image will have nearest neighbours belonging to (mostly) a single class. This information is leveraged by the EnWeDi method to distinguish between the ID and OOD image examples. Using this uncertainty information (quantified as entropy) and combining it with the distance to the nearest neighbour, the EnWeDi method further extends the separation between the ID scores from the OOD scores.



(A) case: in-distribution

(B) case: out-of-distribution

FIGURE 6.6: The number of classes in k-nearest neighbours for ID and OOD images. A set of 400 random images each from the CUB80 test set and CUB120 dataset are taken as ID and OOD

The hypothesis about the uncertainty can be validated by Figure 6.6, which shows the number of classes the nearest neighbours belong to in the feature space for a set of 400 randomly chosen ID and OOD images. The images from the CUB80 dataset and CUB120 dataset are taken as ID and OOD respectively. For the kNN search, the value of k is set at 25, which is half of the average number of images per class in the CUB80 training dataset. As can be seen, around 85% of the ID images have nearest neighbours belonging to a single class, while for the OOD images, only 28% of the images have nearest neighbours belonging to a single class.

The next subsection illustrates how the EnWeDi method improves the separation between ID and OOD in two cases specific to near-OOD detection.

6.4.2 Example Cases of the EnWeDi Method Outperforming the DkNN Method

Case 1: When distances to the nearest neighbours for ID and OOD are similar
In a fine-grained novel-class detection, the near-OOD samples are visually highly similar to the ID classes. Hence the distance to the k^{th} nearest neighbour for an OOD image is at times comparable to that of an ID image, making it hard to set a decision threshold.

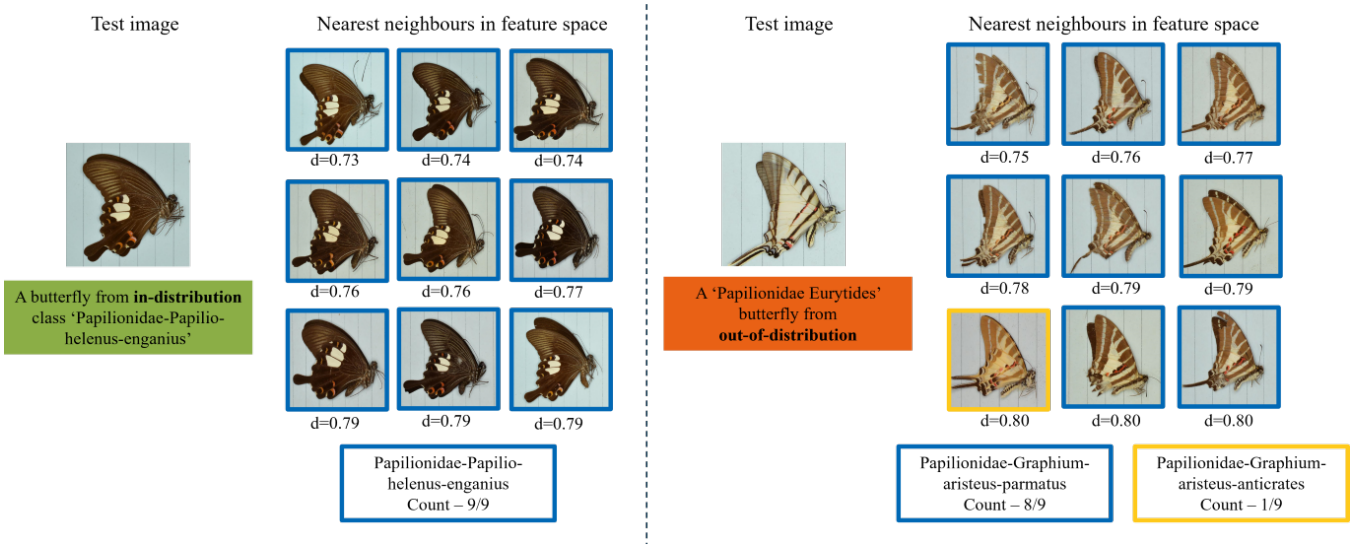


FIGURE 6.7: The case of an ID and OOD image being at similar distances to their respective nearest neighbours in the feature space. The d underneath each nearest neighbour is the Euclidean distance between that neighbour and the test image.

An example is shown in Figure 6.7. In this example, the distances to the k nearest neighbours with $k = 9$ for the ID test image and OOD test image are highly similar. With the DkNN method, the OOD scores would be the distance to the k^{th} nearest neighbour and hence the ID image and the OOD image would get an OOD score of 0.79 and 0.80 respectively. In this case, unless the threshold for the OOD detection was pre-determined to be in between the narrow range of 0.79 and 0.80, the DkNN method would classify both images as belonging to the same distribution (either as ID or OOD). However, for the OOD image, even though the distances to the k-nearest neighbours are similar to that of the ID image, it is observed that the k-nearest neighbours do not belong to the same class. By quantifying this uncertainty and combining it with the distance to the nearest neighbour, the EnWeDi method assigns the OOD score of 0.73 and 0.89 for the ID and OOD image

respectively, making it easier to distinguish between them.

Case 2: When distances to the nearest neighbours for an OOD image are smaller than that of an ID image It has also been observed that there are instances of the ID image's distance to its k^{th} nearest neighbour being greater than that of a near-OOD image. One such example is shown in Figure 6.8 from the near-OOD detection case on the CUB80 dataset. In such cases, the DkNN method, which uses the distance to the k^{th} nearest neighbour, performs terribly by classifying the ID image as OOD and/or the OOD image as ID. The DkNN method would assign ID and OOD images the OOD score of 0.78 and 0.67 respectively (distances to the 9^{th} nearest neighbour). Whereas, the EnWeDi method would assign scores of 0.64 and 0.96 for the ID image and the OOD image respectively, hence increasing the chance of classifying them correctly.

6.4.3 Explainable OOD Detection

Apart from the improved OOD detection capability, the EnWeDi (as well as DkNN) method can also provide an 'explainable OOD score' - an aspect of OOD detection that has been quite overlooked in the literature. Since the EnWeDi method uses labelled feature embeddings of the reference ID dataset, it can provide better insights into why an input image is classified as ID or OOD when compared with other methods in this work.

For instance, as shown in Figure 6.8, for the OOD input image, the OOD score can be explained in terms of the 3 classes it is most visually similar. Furthermore, although the EnWeDi method uses only the distance to the nearest neighbour for calculating the OOD score, a kNN search anyway results in the distances for the k-nearest neighbours. These distances can be further used to understand how closely an input image resembles a particular ID class. If the reference feature embeddings are indexed i.e., they can be traced to the images to which they belong, then an OOD score of an input can be traced to the reference set of images as well.

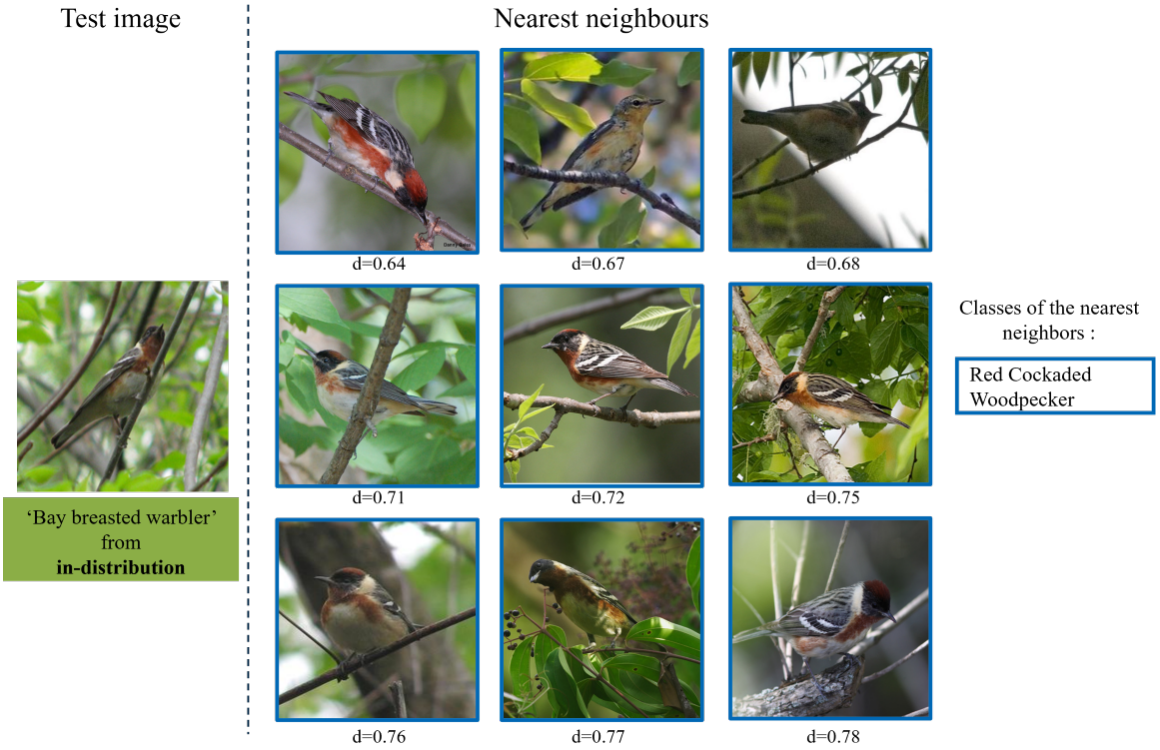
Note that the DkNN method can also give most of the insights mentioned above if the required information (image labels, feature embeddings-image mapping) is available. However, if this information is available, one might as well use the EnWeDi method instead of the DkNN method for improved OOD detection.

6.4.4 Limitations of the EnWeDi Method

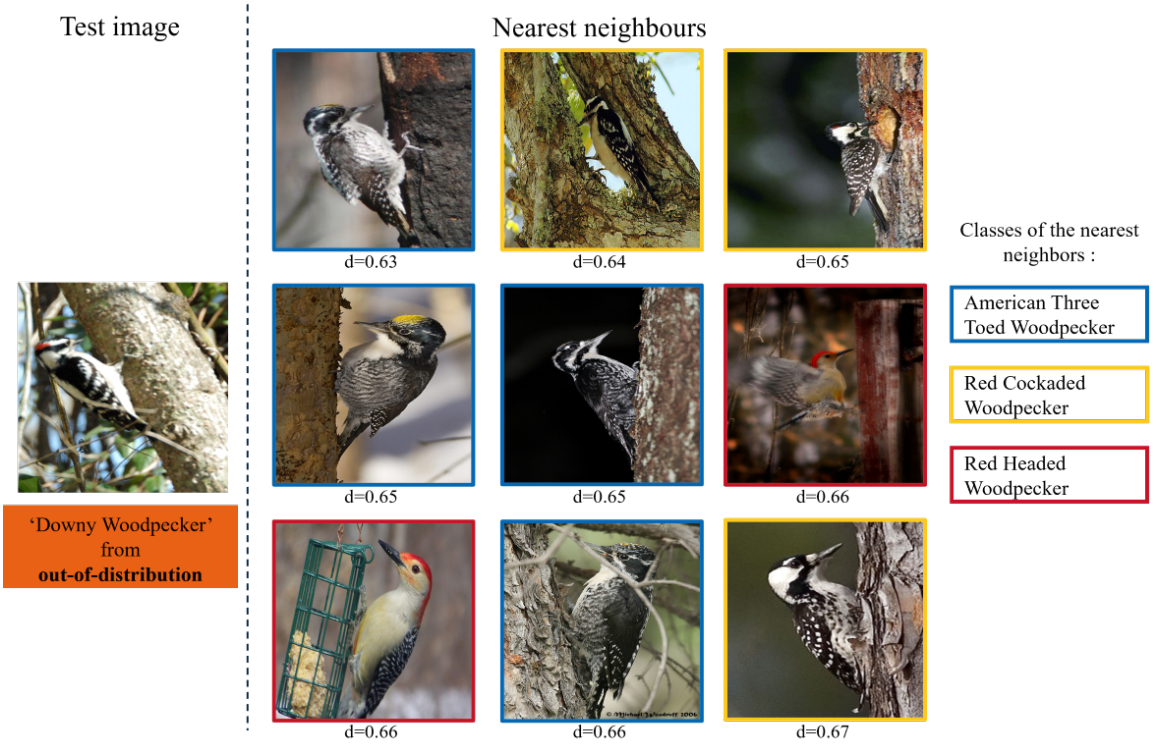
Since EnWeDi depends on the class labels, it can not be applied when labelled training/reference ID images are not available. Furthermore, the EnWeDi method relies on entropy to outperform other methods. Calculation of entropy assumes 'variety' in the inputs, i.e., the number of classes represented in the feature space is more than one. This assumption can not be applied to OOD detection tasks where the number of ID classes is one. For instance, anomaly detection for industrial applications (as seen in the MVTec dataset [103]) could have a single ID class labelled 'Good' or 'Defect-free'. In such cases, the EnWeDi method performs as the DkNN method set with $k = 1$.

6.4.5 Feasibility of the EnWeDi Method

It is to be noted that the EnWeDi method requires performing a k-nearest neighbour (kNN) search on the feature embeddings which is demanding in terms of computations and memory required. Therefore, although it achieves high OOD detection accuracies, it may not be the most ideal method for deploying on edge devices with limited resources



(A) Test image from ID



(B) Test image from OOD

FIGURE 6.8: An example case where the distance to the k^{th} nearest neighbour of an ID image is greater than that of an OOD image

especially when the ID dataset is large. A combination of the below ideas can be used to make the kNN searches and thereby the EnWeDi method more feasible

- **Using fewer feature embeddings** to represent the ID dataset in the feature space. This can be done in multiple ways.
  - **Random stratified sampling** The simplest way is to use feature embeddings of a certain fraction of randomly selected images from each class in the dataset. As shown in Figure 5.8, even with only 10% images from each class, the EnWeDi achieves better OOD detection accuracy than other methods with 100% of the data on the Papilionidae50 dataset. However, on the CUB80 dataset, at least 30% of the dataset was required to achieve the same. It is also to be noted that with 30% of ID data, on both datasets, the EnWeDi method can already attain OOD accuracy that is reasonably close to that when the whole dataset is used.
  - **Coreset sampling** Instead of random sampling, a Coreset selection [104] of the images can be done to select the most informative samples from a class or dataset.
- **Dimensionality reduction of feature embeddings** Despite the high dimensionality, many points in the feature space do not correspond to realistic natural images [5]. This means that the high-dimensional feature embeddings contain sparse information and can be reduced to fewer dimensions without losing much of the information. Therefore principal component analysis (PCA) or similar dimensionality reduction methods can be applied to the feature embeddings to represent them in subspaces of a (fixed) fewer dimensions before adding them to the kNN search space (feature space).

6.4.6 EnWeDi with k Correction

The entropy calculation of the EnWeDi method works best if the number of images per each class (in the feature space) is greater than the value of 'k' chosen. However, when this is not the case, the entropy calculated does not fully represent the measure of uncertainty. An example of this is shown in Figure 6.9. Here, the value of k is set to 9 but the ID input image belongs to a class that has only 2 images in the feature space. These two images are found as the two nearest neighbours for the input image. Hence, intuitively it can be said that the image most likely belongs to ID. The distance to the nearest neighbour (which is used in the calculation of the OOD score) is low and indicates that the image belongs to ID. However, the entropy calculated will indicate that the image might not be belonging to ID.

To overcome this, the 'Entropy weighted nearest neighbour's distance with k-correction' or 'EnWeDiCo' method as an extension of EnWeDi method is proposed. In this method, after a kNN search, the nearest neighbour's class is noted. If all the images belonging to this class are already found in the kNN search, then the entropy is set to 0.

Furthermore, it can also be observed that there is a significant difference in the distance to the 2^{nd} nearest neighbour and the 3^{rd} nearest neighbour. This information can be used in addition to the k-correction.

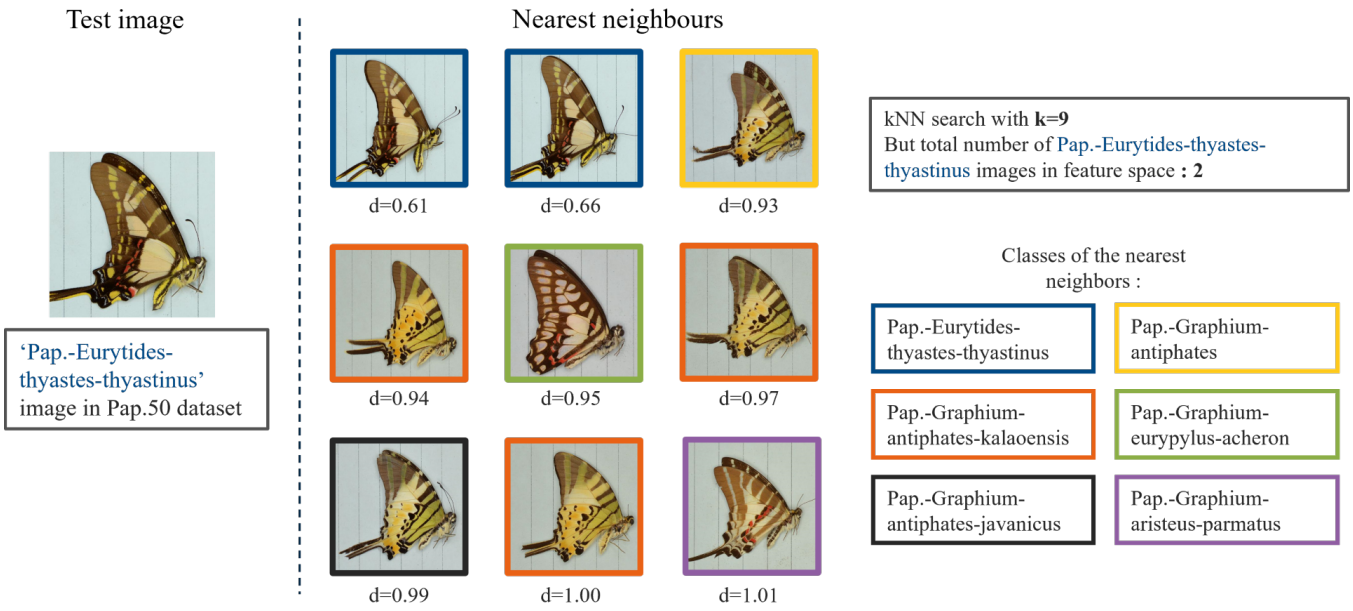


FIGURE 6.9: Example case where the entropy calculation in EnWeDi can be improved. The k hyperparameter for the method is set at 9 while the number of images belonging to the ID class is only two.

Due to time constraints, this method was not completely evaluated in this work. The benefits and limitations of the EnWeDiCo method can be an interesting direction to explore in future.

6.5 To Stack or Not to Stack the Feature Embeddings

In this section, the research question *"What is the effect of stacking feature embeddings from the intermediate layers of a convolutional neural network on the accuracies of the OOD detection methods that make use of feature embeddings?"* is discussed.

As mentioned in Section 4.4, in this work, the feature embeddings-based methods are also tested on the feature embeddings extracted from the intermediate layers and stacked. From the results shown in Section 5.7, it can be seen that this improves the OOD detection capability of the methods when tested on the Papilionidae50 dataset but reduces the same on the CUB80 dataset. In this section, a hypothesis on why this happens and a solution to prevent the degradation of OOD performance seen on the CUB80 dataset is proposed.

6.5.1 What Do the Intermediate Layers Capture?

To understand the effect of stacking of feature embeddings on OOD detection, it is first required to understand what information is extracted from each layer. More specifically, the differences between what each intermediate layers in the CNN represent, specific to the images from the Papilionidae50 and CUB80 datasets are to be understood. This was done by visualising the feature activation maps at each layer from the CNN trained on the respective ID datasets. At each intermediate layer, a feature activation map is extracted by taking the mean along the 'channel' axis on the absolute of the outputs (before the global average pooling layer) from that layer. Figure 6.10 shows these feature activation maps obtained at various intermediate layers for an ID image from the Papilionoidae50 and

CUB80 datasets. As can be seen from the feature activation maps, the deeper layers (for instance, layer 8 and layer 7) capture features more specific to the object of importance in the image. Whereas, the initial layers (for instance, layer 0 to layer 2) extract general features such as edges and global textures.

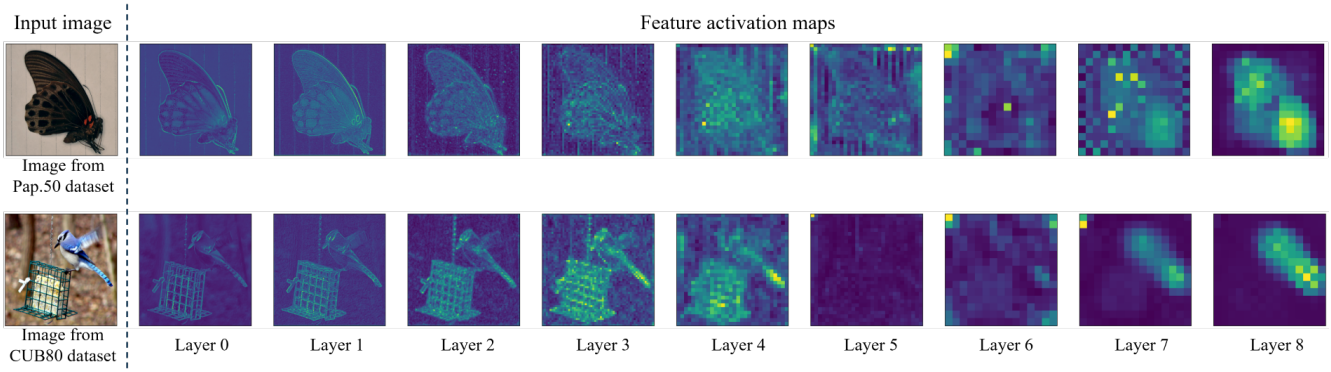


FIGURE 6.10: Feature activation maps using EfficientNet-V2-M trained on respective datasets. Layer 8 is the closest to the output

6.5.2 Why Stacking Features is Not Always Useful for OOD Detection

Through the feature activation maps shown in Figure 6.10, a hypothesis is made to reason why stacking the feature embeddings was beneficial and detrimental for OOD detection on the Papilionidae50 and CUB80 datasets respectively.

A typical image from the CUB80 dataset consists of the subject (the bird) and the background (cage, tree trunks, etc). As it can be seen in Figure 6.10, the feature embeddings extracted from intermediate layers contain a significant amount of information about the background as well. Hence, in a stacked feature embedding vector, a significant number of components contain information belonging to the background. Furthermore, images from OOD, especially from near-OOD (other species of birds) have similar backgrounds as the ID as well. This makes the stacked feature embedding of an ID and OOD image more similar to each other, thereby reducing the OOD detection capability of the methods.

For the Papilionidae50 dataset, the images are taken with subjects (butterflies) on a plain background. This background is almost featureless, i.e., the convolutional layers of the CNN do not extract any significant amount of features from this background. This means that (almost) all the features extracted from intermediate layers belong to the subject itself. Hence, a stacked feature embedding of an image from the Papilionidae50 (and Papilionidae62 OOD) dataset contains more information on the subject when compared to the feature embedding taken only from the penultimate layer (layer 8), thereby leading to an increase in the OOD detection accuracy.

6.5.3 Towards a Solution to Make OOD Detection with Stacked Features Embeddings Robust

In this subsection, a solution to prevent the degradation of OOD performance with stacked feature embeddings is proposed. The solution is based on the idea that using a feature activation map from the penultimate layer, one can find the approximate position of the subject of importance (such as a bird in the CUB80 dataset) and thereby be able to isolate it from the background as much as possible. The process is explained below.

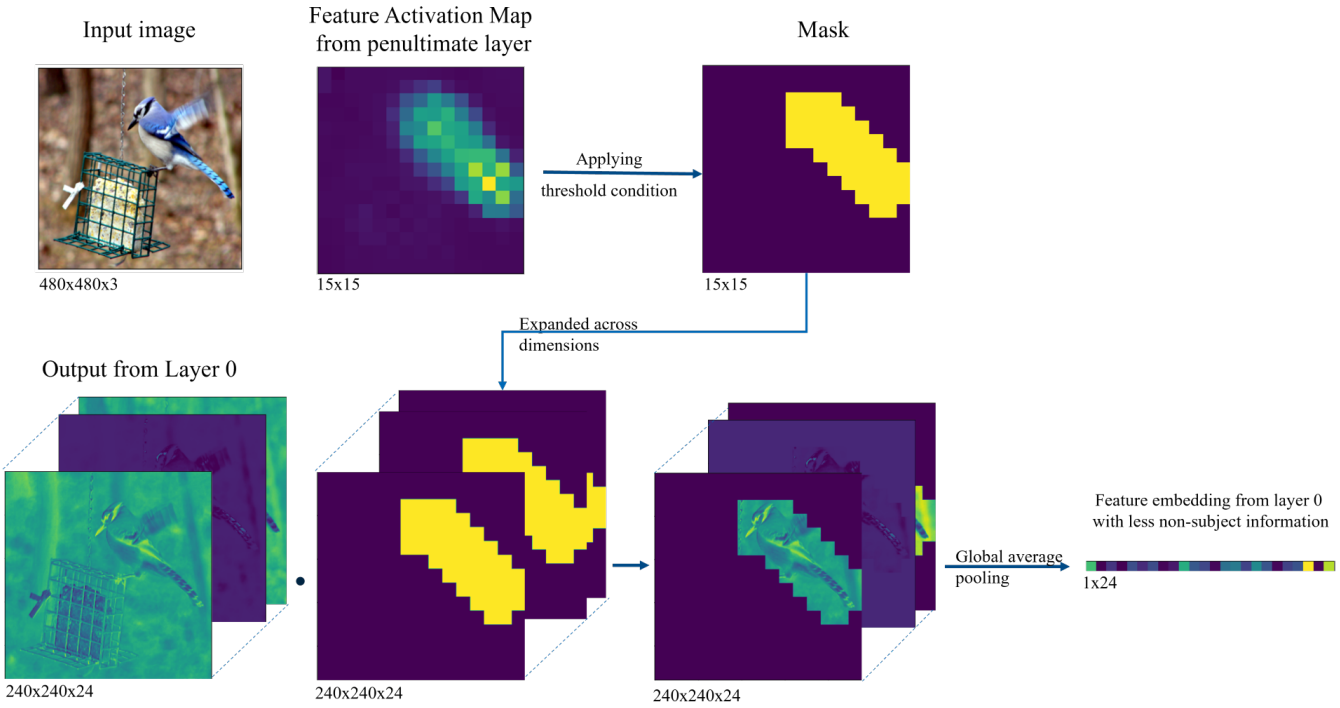


FIGURE 6.11: An example of how AutoCrop is applied to get a feature embedding with non-subject (background) information filtered out

First, a feature activation mask is created by applying a threshold condition on the intensities of the feature activation map obtained from the penultimate layer. A point with an intensity less than the set threshold is marked as 0 (false) in the mask and vice versa. In the mask so obtained, the location of 1s (true) represents the approximate location of the subject. The threshold is a hyperparameter and was set to 30% of the highest intensity value of the feature activation map. This mask is then applied along the channel axis of the feature activation maps obtained from the intermediate layers. Note that the mask is scaled to match the size of the feature activation maps of intermediate layers. The feature embeddings are then extracted by applying the global average pool on the masked feature activation maps. These steps are shown in Figure 6.11. The assumption is that these feature embeddings when stacked would contain considerably less information about the background and hence must be able to prevent degradation of OOD performance. This proposed solution is called 'AutoCrop' feature embeddings in this work.

6.5.4 Results of the Proposed 'AutoCrop' Solution

Figure 6.12 shows the AUROC for near-OOD detection on the CUB80 dataset with and without the proposed AutoCrop solution. With AutoCrop, stacking feature embeddings from shallower layers prevents the huge drop in AUROC for OOD detection, making it a better way to stack the feature embeddings. Furthermore, with AutoCrop there is a small improvement of 0.3% and 0.6% of AUROC on the DkNN and Global FRE PCA method even without stacking the feature embeddings. It has been observed that using AutoCrop on the Papilionidae50 dataset does not show any significant improvement in OOD detection.

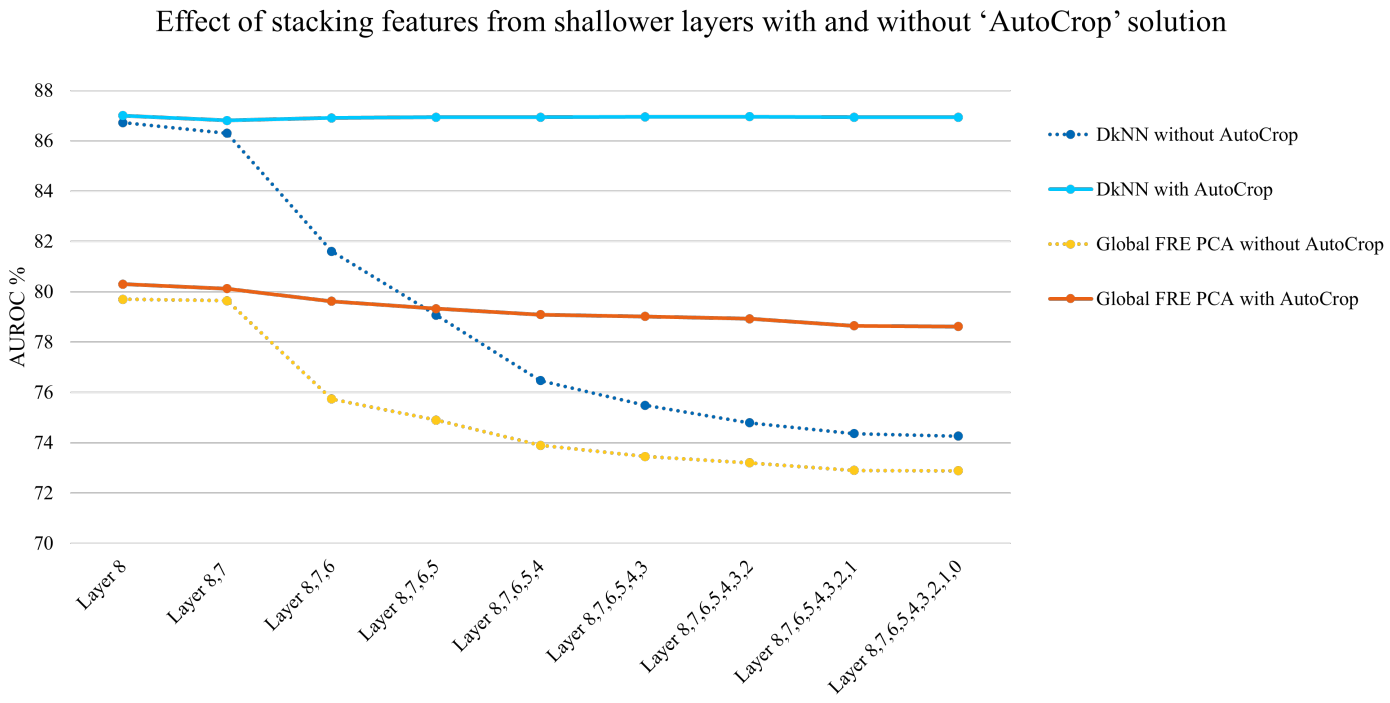


FIGURE 6.12: Results on near-OOD detection for the CUB80 dataset with and without the proposed AutoCrop solution

Overall, AutoCrop is a robust (and safe) way of stacking feature embeddings for the task of OOD detection. Ways to further improve the AutoCrop solution for stacking feature embeddings can be explored in future research.

6.6 Global vs Per-Class Approach to OOD detection Methods

As shown in Section 5.6, the kNN-based and feature reconstruction error (FRE)-based OOD detection methods can be applied 'globally' by considering the whole ID dataset as a single entity or can be applied per each class in the ID dataset. This section provides an analysis and tries to provide a direction on when to use each approach. A few reasons why a particular approach improves the OOD detection accuracy for a particular dataset and decreases the same for another dataset are hypothesised.

For both kNN-based and PCA FRE-based methods, the global approach can be considered as a semi-supervised OOD detection approach and is the only way when the labels of ID images are not available. This could be the case when an OOD detection method needs to be appended to an existing classifier which is already trained. However, applying these methods per class have a few advantages as well.

In the case of the kNN-based OOD detection methods, using a per-class method i.e, Centroid-DkNN reduces the number of feature embeddings in the feature space. For instance, the reduction is about 47x for the CUB80 with 80 ID classes and 108x for the iNaturalist2100 dataset with 2100 ID classes. The increase in kNN search speed due to this depends on the hardware used (as shown in Section 5.5). In general, this results in a faster kNN search which can be performed with fewer computations (and therefore less

energy) and a smaller memory requirement. This makes the Centroid-DkNN a better alternative to the DkNN method for deploying on-edge and low-power devices, especially for large datasets. However, as seen in Figure 5.12 applying Centroid-DkNN lead to a decrease in AUROC for the near-OOD detection task on the long-tailed Papilionidae50 dataset.

In general, a larger number of feature embeddings are required to make a well-representative and robust centroid for a class of images. The centroid made from a few images does not generalise well and is an under-representation of the class.

In the long-tailed Papilionidae50 dataset, the number of images per class is quite low for most of the classes. For instance, in the training split of the dataset, 70% of the classes (35 out of 50) have less than 10 images and 50% of the classes have less than 5 images. The centroids created for these classes would not be accurate representations for the respective classes and hence could be the reason behind the decrease in the OOD detection. Hence, for better OOD detection accuracy, it is recommended to use the per-class kNN-based method only when there are enough images per class to get close representations of the classes. The optimal number of feature embeddings required for a given dataset depends on various factors, including the number of images in the dataset, the complexity of the dataset, and the number of dimensions in the embeddings. On the other extreme, if too many images are used for creating the centroid, the centroid could be too specific to the images of the training set and an overfit representation of the class which might not generalise well to other ID images of the same class.

For the PCA FRE method, when compared with the global approach, the per-class approach led to a decrease (of 4%) in the overall AUROC for near-OOD detection task on the long-tailed Papilionidae50 dataset but lead to a significant increase (>8%) for the same on the CUB80 dataset. In the global approach, the whole dataset is considered as a single entity which might not adequately model the feature spaces underlying structures. The per-class approach takes advantage of the multiple well-separated clusters in the feature space, each of which corresponds to a separate class. Modelling the feature subspaces separately for each class often results in much better OOD performance [5] as shown by the results on the CUB80 dataset.

However, the per-class approach would require more computations and memory when compared to the Global approach. This is because, instead of a single principal component analysis (PCA) model for the entire dataset, multiple PCA models, one for each class, need to be trained (which is done offline), stored and executed for inference. A PCA model essentially consists of Eigen values and Eigen vectors of the covariance matrix of dimensionality reduction. Performing dimensionality reduction and reconstruction of the feature embedding involves computationally expensive and power-hungry matrix multiplications.

On the contrary, on the Papilionidae50 dataset, using the per-class approach does not increase OOD performance. The reason is again tied to the number of images per class in the Papilionidae50 dataset. As principal component analysis (PCA) is based on the correlation between the variables (dimensions) of feature embeddings, to derive a reliable PCA, a considerable number of feature embeddings (ID images) is required. As mentioned earlier, 50% of the classes have less than 5 images per class which question the accuracy and reliability of the PCA models trained on those classes. Furthermore, in [5] the authors also suggest using the Global PCA FRE method when the number of images per class is relatively small to the dimensions of feature embeddings (1280 in this case).

In general, when the dataset contains many small classes, which is a common trait in biodiversity datasets, it is recommended to use the global method over the per-class methods.

Approach to check the suitability of Per-class PCA FRE method on a given dataset There are multiple rules of thumb laid out on the minimum number of samples required for PCA. However, the authors in [105] show that the rules of thumb are not reliable and that the minimum number of samples required for a reliable PCA depends on various factors. These factors include the number of dimensions of the samples and the extent to which the set of principal components can explain the variance inherent in the samples. This is also reflected in the results shown in Figure 5.12. Therefore, the following heuristic is proposed to know if there are enough images per class to follow the per-class approach. One of the signs of an unreliable PCA model is instability. This can be found out by cross-validation (bootstrap) on the PCA model :

1. Choose the class with the least number of images in the ID dataset.
2. Choose 90% of the images from that class and extract feature embeddings for these images.
3. Create a PCA model with a fixed amount of variance to be retained.
4. Calculate the average feature reconstruction error (FRE) on the remaining 10% of the images.
5. Repeat steps 2-4 multiple times with different sets of images in each iteration.
6. If the deviation in the average FREs obtained by different PCA models is within an acceptable range, then the per-class approach can be followed.

6.7 Deploying OOD Detection Methods on Low-Power Edge Devices

While this work has focused on analysing existing and developing new OOD detection methods for biodiversity datasets, it is also important to discuss the practicality of deploying these methods on low-power edge devices. The following points can be used as starting points for research on deploying OOD detection methods on low-power edge devices.

- **Investigating the impact of model compression and quantisation techniques on OOD detection accuracy**. Model compression and quantisation techniques can significantly reduce the size and computational requirements of the classifier. But this also has an impact on classification accuracy. Future research could explore how different OOD detection methods perform in terms of accuracy and speed on quantised models. A design methodology to deploy OOD detection methods using quantised models is presented in [106].
- **Optimised OOD detection methods based on the target hardware** Future research could explore the development of novel OOD detection methods or optimise existing OOD detection methods for deploying on hardware with limited resources. For instance, the trade-offs between speed, memory requirement and accuracy of various fast and approximate kNN search algorithms can be tested specifically on the hardware on which the kNN-based OOD detection method will be deployed. OOD detection methods that take advantage of the existing hardware accelerators on the target device can be researched.

- **Power Consumption of OOD Detection Methods** While many existing OOD detection methods have been developed for high-powered computing environments, there is a need for new methods that are specifically designed for low-power devices. Power consumption of the OOD detection methods is still very limited research. In [107], the authors optimise an anomaly detection solution and estimate its power consumption on an FPGA.

- **Model Splitting for OOD detection** An interesting approach to OOD detection to save computations is EARLIN (EARLy OOD detection for Collaborative INference)[108]. EARLIN partitions a trained model (using model splitting [109]) and deploys only a few shallow layers of it on the edge device and the rest on the cloud. The smaller partition of the model is used to get an initial estimate of the OODness of the input image. This estimate is used to decide whether to upload the image for further processing on the cloud.

- **Pre-Inference OOD Detection** On low-power edge devices, usually the image is captured on the edge device and is uploaded to the cloud where the heavy deep learning model inferencing is done. However, uploading an image is quite power-hungry, hence if the edge device could detect or estimate to a reasonable degree that an input image is OOD, it could potentially save power by not uploading that input to the cloud. However, inferencing an image on deep neural networks is also computationally expensive. Therefore if possible, approximate OOD detection methods that could detect OOD images (at least far-OOD images) as OOD without using neural networks can be explored. For instance, the method presented in [110] makes use of features extracted from scale-invariant feature transform (SIFT) [111] to perform OOD detection. Although SIFT features might not perform as well as the features extracted from a CNN, in [112] the authors show that, in terms of the computation cost and the time taken, SIFT is better than CNNs. Similarly, the authors in [113] propose a method that extracts visual and textural features from an image using classic image processing algorithms such as Haralick features [114] to detect previously unseen images. The key to identifying new scenes was based on a co-occurrence matrix which is a lookup table containing information about which species appear in which environmental conditions.

6.8 Limitations

Beyond accuracy and inference time of OOD detection methods The OOD detection methods in this work are not compared in terms of the memory required to execute them. OOD detection methods that require a large amount of memory may not be feasible to use in resource-constrained environments, such as on mobile devices or in real-time systems.

OOD detection on various CNN architectures The EfficientNetV2-M [70] CNN model was used in all the experiments in this work. The choice is motivated by reasons mentioned in Subsection 4.3.1. Trying OOD detection methods on CNN models of different sizes and different architectures could give insights into the consistency of OOD detection capabilities of the method across those models. This would have also been useful in understanding if a particular OOD detection method is better suited for a particular type of CNN architecture.

Reduced number of experiments on the iNaturalist2100 dataset With more than 0.2 million images, the iNaturalist2100 dataset posed a challenge in terms of the time taken for each experiment. For the case of near-OOD detection, the number of images to be tested is around 0.5 million images. Hence, experiments such as the effect of stacking feature embeddings or the effect of using a reduced amount of data on the OOD detection are performed with a reduced set of hyperparameters on the iNaturalist2100, to corroborate the results shown on the other two datasets. Anomaly detection on iNaturalist2100 is also not tested for the same reasons.

Measuring fine-grainedness of the dataset In this work, the fine-grainedness of the datasets is not measured. The fine-grainedness of the dataset could be measured either by extending the concepts used in domain similarity or by using other granularity measurements such as [115]. Measuring the fine-grainedness of a dataset before and after splitting into ID and OOD for testing could be a better way to categorise the OOD test case into near-, intermediate- or far-OOD.

Chapter 7

Conclusions and Future Work

This chapter provides the conclusion of the presented research as well as provides directions of possible future research.

7.1 Conclusion

The problem of out-of-distribution (OOD) detection has gained significant attention in recent years due to the increasing need of employing computer vision models in real-world applications. Making computer vision models 'open-world ready' is crucial for ensuring the reliability and safety of the systems which rely on these models.

Despite the growing interest in OOD detection, OOD detection for long-tailed and fine-grained datasets, such as biodiversity datasets, has not received much attention in the literature and therefore the performance of existing OOD detection methods on such datasets is still largely unknown. This is important as these datasets more closely resemble the distribution of objects in the real/open world, where certain classes are more abundant than others, and rare or unseen classes may occur.

To bridge this gap, the question that this research aims to answer is: ***"For in-distribution datasets, with fine-grained and long-tailed characteristics, what is the best retrofittable out-of-distribution detection method that achieves high accuracy for the task of OOD detection across OOD datasets with varying difficulty?"***

To conclude this research, the research question is addressed by answering the sub-questions.

RQ1: "For a given in-distribution dataset, how can the difficulty of detecting OOD images from a particular OOD dataset be quantified?" Previous works have either neglected to measure the OOD difficulty or have just presumed it subjectively. In this work, the domain similarity score has been used as a way to measure the OODness of a dataset. It was shown that these scores mostly align with the supposition that a lower domain similarity score between two datasets implies the OOD detection between them is more difficult. The domain similarity score is also less dependent on the choice of CNN architecture. However, there are a few instances in the anomaly datasets for which the domain similarity scores do not agree with the OOD detection scores or the intuition of OOD detection difficulty. For instance, for a given image, a new image generated by adding 'motion blur' is intuitively more similar than that for an image with 'coarse dropout' corruption applied. The OOD scores also indicate the same. However, the domain

similarity scores calculated on these sets of images do not conform to this. This leaves for further investigation and improvement in quantifying the OOD detection difficulty.

RQ2: "What improvements can be made to the existing OOD detection methods to further improve their accuracy for the task of OOD detection on fine–grained and long-tailed in-distribution datasets?" The limitations of the underlying hypothesis of the DkNN method (which already achieved relatively high OOD detection accuracies) were identified in this research. These limitations are more applicable when the OOD detection task is between a fine-grained ID dataset and a near-OOD dataset. For instance, the DkNN method relies entirely on the distance to the k-nearest neighbour for OOD detection. It was observed that the distance to the k-nearest neighbour in feature space is not always higher for an OOD image than an ID image. In such cases, the DkNN method fails to distinguish between ID and OOD images. The proposed EnWeDi method overcomes these limitations by using entropy weighted nearest neighbours distance. The EnWeDi achieves higher OOD detection accuracy than DkNN in 10 out of 14 cases and performs similarly to the DkNN method in 3 out of 14 cases. In only one experimental setup the DkNN method performs better than the EnWeDi method - the case of achieving a high TPR at a low FPR on the long-tailed Papilionidae50 dataset. Finally, the inference time of the EnWeDi method is not more than 0.3 ms than that of the DkNN method.

RQ3: "What is the effect of stacking feature embeddings from the intermediate layers of a convolutional neural network on the accuracies of the OOD detection methods that make use of feature embeddings?" It can be inferred that the effect of stacking feature embeddings from intermediate layers increases the OOD detection accuracies significantly if the images in the datasets contain the subject of classification and an almost featureless background. The increase depends on the method and ranges from 2.7% to 6.7% in AUROC.

In cases where a significant portion of the images is background, the stacking of features degrades the OOD detection accuracy (the highest drop in AUROC was 12.3%). For such cases, 'AutoCrop' - a safe and reliable approach to stacking is proposed. The Autocrop method attempts to remove the background from an image so the feature embeddings extracted from the image belong (more exclusively) to the subject. Using Autocrop on such datasets prevented the degradation in OOD performance and lead to a small ($<$1% in AUROC) improvement in OOD detection. The benefits of Autocrop on OOD detection on other datasets is a promising direction for future work.

RQ4: "What are the OOD detection methods that can consistently achieve the best accuracies over different OOD difficulties (from RQ1) when measured across a range of evaluation metrics?" Based on the findings from this research, it can be concluded that the proposed method EnWeDi consistently outperforms other OOD detection methods in terms of accuracy across different OOD difficulties and evaluation metrics. It outperforms all the considered methods in 5 out of 14 cases and performs similarly ($<$1% difference in the corresponding evaluation metric) to the best performing method in the remaining 7 out of 9 cases. Methods that outperform the EnWeDi in a few cases do so in only those cases and show worse performance in other cases or are too slow. For instance, the ODIN method outperforms the EnWeDi method by 0.1% (AUROC) for near-OOD case on the iNaturalist2100 dataset but shows 6% worse AUROC for near-OOD case on the Papilionidae50 dataset. The ODIN method is also at least 5 times slower than the EnWeDi method. These results suggest that EnWeDi is a promising

method for OOD detection. EnWeDi method (just as any other OOD detection method) has to be validated on more datasets to identify any opportunities for further improvement.

7.2 Future Works

"There is more to see than can ever be seen." Despite significant progress in OOD detection, there remains a need for further research to improve the accuracy, efficiency, and applicability of existing methods in real-world scenarios.

Training techniques that improve OOD detection accuracy Training techniques that can be used during the training phase of the classifier to increase the OOD detection performance can be explored. For instance, it has been shown in [6] that training a model with supervised contrastive learning (SupCon) [116] increases the OOD detection capability of the DkNN method. The authors of the Energy method [2] also show that using the Energy function (Equation 3.2) as a loss function for training the classifier further improves the OOD accuracy.

It has been shown in [117], that using Mixup [63] augmentation can improve calibration and predictive uncertainty in neural networks. Similarly, data augmentation techniques such as CutMix [118], Cutout[119] and AugMix [120] are useful in regularising a neural network and providing less over-confident prediction probabilities. Such data augmentation techniques can be explored for improving the OOD detection capabilities of the Baseline MSP or other score-based detection methods.

Most prediction probabilities-based OOD detection methods in this work and the literature rely on assumption that the prediction probabilities are not over-confident. Apart from the temperature scaling [36], techniques such as logit normalisation [121] can be used to reduce the neural network overconfidence, which might improve the OOD detection accuracy.

Deploying OOD detection methods on edge devices Deploying OOD-detection methods on low-power edge devices is a common requirement in a biodiversity setting. As a future work, optimising OOD detection methods to run on given constraints as well as their performance on smaller architectures and quantised model can be analysed. Designing efficient OOD detection methods in combination with techniques such as model-splitting [109] can be further researched. Multiple ideas on deploying the OOD detection methods are also discussed in Section 6.7.

OOD detection beyond image classification task In this work, OOD detection methods are analysed for the task of image classification. However, the underlying concepts of these methods can be extended to be used for OOD detection for other computer vision tasks such as multi-label classification, object detection and semantic segmentation. For instance, an 'unknown-aware' object detection method is presented in [13] and is compared with the Energy, Baseline MSP and ODIN methods adapted for the task of the object detection. Another OOD detection method for the task of object detection is presented in [122]. Similarly, [123] and [124] present OOD detection methods for image segmentation tasks.

Exploring these areas of research can pave the way for more robust OOD detection methods which can improve the safety and reliability of artificial intelligence systems by enabling them to 'know the unknown'.

References

[1] Dan Hendrycks and Kevin Gimpel. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *CoRR* abs/1610.02136 (2016). arXiv: 1610.02136. URL: http://arxiv.org/abs/1610.02136.

[2] Weitang Liu et al. "Energy-based Out-of-distribution Detection". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 21464–21475. URL: https://proceedings.neurips.cc/paper/2020/file/f5496252609c43eb8a3d147ab9b9c006-Paper.pdf.

[3] Shiyu Liang, Yixuan Li, and R. Srikant. *Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks*. 2017. DOI: 10.48550/ARXIV.1706.02690. URL: https://arxiv.org/abs/1706.02690.

[4] Rui Huang, Andrew Geng, and Yixuan Li. "On the Importance of Gradients for Detecting Distributional Shifts in the Wild". In: *CoRR* abs/2110.00218 (2021). arXiv: 2110.00218. URL: https://arxiv.org/abs/2110.00218.

[5] Ibrahima J. Ndiour, Nilesh A. Ahuja, and Omesh Tickoo. *Subspace Modeling for Fast Out-Of-Distribution and Anomaly Detection*. 2022. DOI: 10.48550/ARXIV.2203.10422. URL: https://arxiv.org/abs/2203.10422.

[6] Yiyou Sun et al. *Out-of-Distribution Detection with Deep Nearest Neighbors*. 2022. DOI: 10.48550/ARXIV.2204.06507. URL: https://arxiv.org/abs/2204.06507.

[7] Erich Schubert, Arthur Zimek, and Hans-Peter Kriegel. "Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection." In: *Data Min. Knowl. Discov.* 28.1 (2014), pp. 190–237. URL: http://dblp.uni-trier.de/db/journals/datamine/datamine28.html#SchubertZK14.

[8] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *CoRR* abs/1409.0575 (2014). arXiv: 1409.0575. URL: http://arxiv.org/abs/1409.0575.

[9] *Image Classification on ImageNet - Papers with code*. https://paperswithcode.com/sota/image-classification-on-imagenet. Accessed: 12 September, 2022.

[10] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. "Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images". In: *CoRR* abs/1412.1897 (2014). arXiv: 1412.1897. URL: http://arxiv.org/abs/1412.1897.

[11] Christian Szegedy et al. "Intriguing properties of neural networks". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. URL: http://arxiv.org/abs/1312.6199.

[12] Seyed-Mohsen Moosavi-Dezfooli et al. “Universal adversarial perturbations”. In: *CoRR* abs/1610.08401 (2016). arXiv: 1610.08401. URL: http://arxiv.org/abs/1610.08401.

[13] Xuefeng Du et al. *Unknown-Aware Object Detection: Learning What You Don't Know from Videos in the Wild*. 2022. DOI: 10.48550/ARXIV.2203.03800. URL: https://arxiv.org/abs/2203.03800.

[14] Xiang Sean Zhou and T.S. Huang. “Small sample learning during multimedia retrieval using BiasMap”. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990450.

[15] Ziwei Liu et al. “Large-Scale Long-Tailed Recognition in an Open World”. In: *CoRR* abs/1904.05160 (2019). arXiv: 1904.05160. URL: http://arxiv.org/abs/1904.05160.

[16] Abhijit Bendale and Terrance E. Boult. “Towards Open World Recognition”. In: *CoRR* abs/1412.5687 (2014). arXiv: 1412.5687. URL: http://arxiv.org/abs/1412.5687.

[17] Si Liu et al. “Open Category Detection with PAC Guarantees”. In: *CoRR* abs/1808.00529 (2018). arXiv: 1808.00529. URL: http://arxiv.org/abs/1808.00529.

[18] Zhen Fang et al. “Learning Bounds for Open-Set Learning”. In: *CoRR* abs/2106.15792 (2021). arXiv: 2106.15792. URL: https://arxiv.org/abs/2106.15792.

[19] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly Detection: A Survey”. In: *ACM Comput. Surv.* 41.3 (July 2009). ISSN: 0360-0300. DOI: 10.1145/1541880.1541882. URL: https://doi.org/10.1145/1541880.1541882.

[20] Jingkang Yang et al. “Generalized Out-of-Distribution Detection: A Survey”. In: *CoRR* abs/2110.11334 (2021). arXiv: 2110.11334. URL: https://arxiv.org/abs/2110.11334.

[21] Victoria Hodge and Jim Austin. “A Survey of Outlier Detection Methodologies”. In: *Artificial Intelligence Review* 22 (Oct. 2004), pp. 85–126. DOI: 10.1023/B:AIRE.0000045502.10941.a9.

[22] Corinna Cortes and Vladimir Naumovich Vapnik. “Support-Vector Networks”. In: *Machine Learning* 20 (2004), pp. 273–297.

[23] Walter J. Scheirer et al. “Toward Open Set Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.7 (2013), pp. 1757–1772. DOI: 10.1109/TPAMI.2012.256.

[24] Hakan Cevikalp. “Best Fitting Hyperplanes for Classification”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pp. 1076–1088. DOI: 10.1109/TPAMI.2016.2587647.

[25] Walter J. Scheirer, Lalit P. Jain, and Terrance E. Boult. “Probability Models for Open Set Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.11 (2014), pp. 2317–2324. DOI: 10.1109/TPAMI.2014.2321392.

[26] Lalit Jain, Walter Scheirer, and Terrance Boult. “Multi-class Open Set Recognition Using Probability of Inclusion”. In: vol. 8691. Sept. 2014, pp. 393–409. ISBN: 978-3-319-10577-2. DOI: 10.1007/978-3-319-10578-9_26.

[27] S. Kotz and S. Nadarajah. *Extreme Value Distributions*. World Scientific Publishing Company, 2000. ISBN: 9781783261734. URL: https://books.google.nl/books?id=ZPW3CgAAQBAJ.

[28] Matthew D. Scherreik and Brian D. Rigling. "Open set recognition for automatic target classification with rejection". In: *IEEE Transactions on Aerospace and Electronic Systems* 52.2 (2016), pp. 632–642. DOI: 10.1109/TAES.2015.150027.

[29] Thomas Mensink et al. "Distance-Based Image Classification: Generalizing to New Classes at Near-Zero Cost". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.11 (2013), pp. 2624–2637. DOI: 10.1109/TPAMI.2013.83.

[30] Pedro Ribeiro Mendes Júnior et al. "Nearest neighbors distance ratio open-set classifier". In: *Machine Learning* 106 (2016), pp. 359–386.

[31] Markus M. Breunig et al. "LOF: Identifying Density-Based Local Outliers". In: *SIGMOD Rec.* 29.2 (May 2000), pp. 93–104. ISSN: 0163-5808. DOI: 10.1145/335191.335388. URL: https://doi.org/10.1145/335191.335388.

[32] Ke Zhang, Marcus Hutter, and Huidong Jin. "A New Local Distance-Based Outlier Detection Approach for Scattered Real-World Data". In: *CoRR* abs/0903.3257 (2009). arXiv: 0903.3257. URL: http://arxiv.org/abs/0903.3257.

[33] He Zhang and Vishal M. Patel. "Sparse Representation-Based Open Set Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.8 (2017), pp. 1690–1696. DOI: 10.1109/TPAMI.2016.2613924.

[34] Ethan M. Rudd et al. "The Extreme Value Machine". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.3 (Mar. 2018), pp. 762–768. DOI: 10.1109/tpami.2017.2707495. URL: https://doi.org/10.1109%2Ftpami.2017.2707495.

[35] Abhijit Bendale and Terrance E. Boult. "Towards Open Set Deep Networks". In: *CoRR* abs/1511.06233 (2015). arXiv: 1511.06233. URL: http://arxiv.org/abs/1511.06233.

[36] Chuan Guo et al. "On Calibration of Modern Neural Networks". In: *CoRR* abs/1706.04599 (2017). arXiv: 1706.04599. URL: http://arxiv.org/abs/1706.04599.

[37] Yen-Chang Hsu et al. "Generalized ODIN: Detecting Out-of-distribution Image without Learning from Out-of-distribution Data". In: *CoRR* abs/2002.11297 (2020). arXiv: 2002.11297. URL: https://arxiv.org/abs/2002.11297.

[38] Yann LeCun et al. "A Tutorial on Energy-Based Learning". In: 2006.

[39] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. "Deep Anomaly Detection with Outlier Exposure". In: *CoRR* abs/1812.04606 (2018). arXiv: 1812.04606. URL: http://arxiv.org/abs/1812.04606.

[40] Yiyou Sun, Chuan Guo, and Yixuan Li. "ReAct: Out-of-distribution Detection With Rectified Activations". In: *CoRR* abs/2111.12797 (2021). arXiv: 2111.12797. URL: https://arxiv.org/abs/2111.12797.

[41] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86. DOI: 10.1214/aoms/1177729694. URL: https://doi.org/10.1214/aoms/1177729694.

[42] Navid Kardan and Kenneth O. Stanley. "Mitigating fooling with competitive overcomplete output layer neural networks". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017, pp. 518–525. DOI: 10.1109/IJCNN.2017.7965897.

[43] Nicolas Papernot and Patrick D. McDaniel. "Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning". In: *CoRR* abs/1803.04765 (2018). arXiv: 1803.04765. URL: http://arxiv.org/abs/1803.04765.

[44] Jeff Johnson, Matthijs Douze, and Hervé Jégou. "Billion-scale similarity search with GPUs". In: *IEEE Transactions on Big Data* 7.3 (2019), pp. 535–547.

[45] Ryota Yoshihashi et al. "Classification-Reconstruction Learning for Open-Set Recognition". In: *CoRR* abs/1812.04246 (2018). arXiv: 1812.04246. URL: http://arxiv.org/abs/1812.04246.

[46] Seiya Tokui et al. "Chainer: A Deep Learning Framework for Accelerating the Research Cycle". In: *CoRR* abs/1908.00213 (2019). arXiv: 1908.00213. URL: http://arxiv.org/abs/1908.00213.

[47] Rui Huang and Yixuan Li. "MOS: Towards Scaling Out-of-distribution Detection for Large Semantic Space". In: *CoRR* abs/2105.01879 (2021). arXiv: 2105.01879. URL: https://arxiv.org/abs/2105.01879.

[48] Douglas O. Cardoso, Felipe França, and João Gama. "A bounded neural network for open set recognition". In: *2015 International Joint Conference on Neural Networks (IJCNN)*. 2015, pp. 1–7. DOI: 10.1109/IJCNN.2015.7280680.

[49] Igor Aleksander et al. "A brief introduction to Weightless Neural Systems". In: Jan. 2009.

[50] Yifei Ming et al. *CIDER: Exploiting Hyperspherical Embeddings for Out-of-Distribution Detection*. 2022. DOI: 10.48550/ARXIV.2203.04450. URL: https://arxiv.org/abs/2203.04450.

[51] Poojan Oza and Vishal M. Patel. "C2AE: Class Conditioned Auto-Encoder for Open-set Recognition". In: *CoRR* abs/1904.01198 (2019). arXiv: 1904.01198. URL: http://arxiv.org/abs/1904.01198.

[52] Dan Hendrycks et al. "Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty". In: *CoRR* abs/1906.12340 (2019). arXiv: 1906.12340. URL: http://arxiv.org/abs/1906.12340.

[53] Yifei Ming, Ying Fan, and Yixuan Li. "POEM: Out-of-Distribution Detection with Posterior Sampling". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, July 2022, pp. 15650–15665. URL: https://proceedings.mlr.press/v162/ming22a.html.

[54] Xiaolong Wang et al. "Non-local Neural Networks". In: *CoRR* abs/1711.07971 (2017). arXiv: 1711.07971. URL: http://arxiv.org/abs/1711.07971.

[55] ZongYuan Ge et al. "Generative OpenMax for Multi-Class Open Set Classification". In: *CoRR* abs/1707.07418 (2017). arXiv: 1707.07418. URL: http://arxiv.org/abs/1707.07418.

[56] Ian Goodfellow et al. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[57] Yann LeCun, Corinna Cortes, and CJ Burges. "MNIST handwritten digit database". In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2 (2010).

[58] Martin Thoma. "The HASYv2 dataset". In: *CoRR* abs/1701.08380 (2017). arXiv: 1701.08380. URL: http://arxiv.org/abs/1701.08380.

[59] Lawrence Neal et al. “Open Set Learning with Counterfactual Images”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.

[60] Inhyuk Jo et al. “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)*. 2018, pp. 2686–2690. DOI: 10.1109/ICASSP.2018.8461700.

[61] Shu Kong and Deva Ramanan. “OpenGAN: Open-Set Recognition via Open Data Generation”. In: *CoRR* abs/2104.02939 (2021). arXiv: 2104.02939. URL: https://arxiv.org/abs/2104.02939.

[62] Ryne Roady, Tyler L. Hayes, and Christopher Kanan. “Improved Robustness to Open Set Inputs via Tempered Mixup”. In: *CoRR* abs/2009.04659 (2020). arXiv: 2009.04659. URL: https://arxiv.org/abs/2009.04659.

[63] Hongyi Zhang et al. “mixup: Beyond Empirical Risk Minimization”. In: *CoRR* abs/1710.09412 (2017). arXiv: 1710.09412. URL: http://arxiv.org/abs/1710.09412.

[64] Sagar Vaze et al. “Open-Set Recognition: A Good Closed-Set Classifier is All You Need”. In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=5hLP5JY9S2d.

[65] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and Harnessing Adversarial Examples”. In: *International Conference on Learning Representations*. 2015. URL: http://arxiv.org/abs/1412.6572.

[66] Jonathon Shlens. “A Tutorial on Principal Component Analysis”. In: *CoRR* abs/1404.1100 (2014). arXiv: 1404.1100. URL: http://arxiv.org/abs/1404.1100.

[67] C. E. Shannon. “A mathematical theory of communication”. In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[68] Yin Cui et al. “Large Scale Fine-Grained Categorization and Domain-Specific Transfer Learning”. In: *CoRR* abs/1806.06193 (2018). arXiv: 1806.06193. URL: http://arxiv.org/abs/1806.06193.

[69] Y. Rubner, C. Tomasi, and L.J. Guibas. “A metric for distributions with applications to image databases”. In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 1998, pp. 59–66. DOI: 10.1109/ICCV.1998.710701.

[70] Mingxing Tan and Quoc V. Le. “EfficientNetV2: Smaller Models and Faster Training”. In: *CoRR* abs/2104.00298 (2021). arXiv: 2104.00298. URL: https://arxiv.org/abs/2104.00298.

[71] Jacob Wilhelm Kamminga. “Hiding in the Deep: Online Animal Activity Recognition using Motion Sensors and Machine Learning”. English. PhD thesis. Netherlands: University of Twente, Sept. 2020. ISBN: 978-90-365-5055-0. DOI: 10.3990/1.9789036550550.

[72] Laurens Hogeweg. “Naturalis Papilionidae Dataset”. In: (Sept. 2021). DOI: 10.6084/m9.figshare.16627324.v1. URL: %5Curl%7Bhttps://figshare.com/articles/figure/figshare_zip/16627324%7D.

[73] C. Wah et al. *Caltech-UCSD Birds-200-2011 (CUB-200-2011)*. Tech. rep. CNS-TR-2011-001. California Institute of Technology, 2011.

[74] Grant Van Horn et al. *The iNaturalist Species Classification and Detection Dataset*. 2017. DOI: 10.48550/ARXIV.1707.06642. URL: https://arxiv.org/abs/1707.06642.

[75] Erik Rodner et al. "Fine-grained Recognition Datasets for Biodiversity Analysis". In: *CoRR* abs/1507.00913 (2015). arXiv: 1507.00913. URL: http://arxiv.org/abs/1507.00913.

[76] Bolei Zhou et al. "Places: A 10 Million Image Database for Scene Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (2018), pp. 1452–1464. DOI: 10.1109/TPAMI.2017.2723009.

[77] Mircea Cimpoi et al. "Describing Textures in the Wild". In: *CoRR* abs/1311.3618 (2013). arXiv: 1311.3618. URL: http://arxiv.org/abs/1311.3618.

[78] Daniel Kermany. *Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification*. eng. 2018.

[79] Irwan Bello et al. "Revisiting ResNets: Improved Training and Scaling Strategies". In: *CoRR* abs/2103.07579 (2021). arXiv: 2103.07579. URL: https://arxiv.org/abs/2103.07579.

[80] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *CoRR* abs/2010.11929 (2020). arXiv: 2010.11929. URL: https://arxiv.org/abs/2010.11929.

[81] S Gupta and M Tan. *EfficientNet-EdgeTPU: Creating Accelerator-Optimized Neural Networks with AutoML*. Accessed: 02 October, 2022. Aug. 2019. URL: https://ai.googleblog.com/2019/08/efficientnet-edgetpu-creating.html.

[82] Amir Yazdanbakhsh et al. "An Evaluation of Edge TPU Accelerators for Convolutional Neural Networks". In: *CoRR* abs/2102.10423 (2021). arXiv: 2102.10423. URL: https://arxiv.org/abs/2102.10423.

[83] *Randomhorizontalflip - Torchvision main documentation*. Accessed: 07 November, 2022. URL: https://pytorch.org/vision/main/generated/torchvision.transforms.RandomHorizontalFlip.html.

[84] *RandomVerticalFlip - Torchvision main documentation*. Accessed: 07 November, 2022. URL: https://pytorch.org/vision/main/generated/torchvision.transforms.RandomVerticalFlip.html.

[85] *RandomRotation - Torchvision main documentation*. Accessed: 07 November, 2022. URL: https://pytorch.org/vision/main/generated/torchvision.transforms.RandomRotation.html.

[86] Sébastien Marcel and Yann Rodriguez. "Torchvision the Machine-Vision Package of Torch". In: *Proceedings of the 18th ACM International Conference on Multimedia*. MM '10. Firenze, Italy: Association for Computing Machinery, 2010, pp. 1485–1488. ISBN: 9781605589336. DOI: 10.1145/1873951.1874254. URL: https://doi.org/10.1145/1873951.1874254.

[87] *CosineAnnealingWarmRestarts - Torch optimization (TORCH.OPTIM)*. Accessed: 20 October, 2022. URL: https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.CosineAnnealingWarmRestarts.html.

[88] Ilya Loshchilov and Frank Hutter. "SGDR: Stochastic Gradient Descent with Restarts". In: *CoRR* abs/1608.03983 (2016). arXiv: 1608.03983. URL: http://arxiv.org/abs/1608.03983.

[89] *ExponentialLR - Torch optimization (TORCH.OPTIM)*. Accessed: 20 October, 2022. URL: https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ExponentialLR.html.

[90] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[91] Rui Huang, Andrew Geng, and Yixuan Li. *On the Importance of Gradients for Detecting Distributional Shifts in the Wild*. https://github.com/deeplearning-wisc/gradnorm_ood. 2021.

[92] Yiyou Sun et al. *Out-of-Distribution Detection with Deep Nearest Neighbors*. https://github.com/deeplearning-wisc/knn-ood. 2022.

[93] Alexander Buslaev et al. "Albumentations: Fast and Flexible Image Augmentations". In: *Information* 11.2 (2020). ISSN: 2078-2489. DOI: 10.3390/info11020125. URL: https://www.mdpi.com/2078-2489/11/2/125.

[94] Dawid Laszuk. *Python implementation of Empirical Mode Decomposition algorithm*. https://github.com/laszukdawid/PyEMD. 2017. DOI: 10.5281/zenodo.5459184.

[95] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[96] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[97] Jesse Davis and Mark Goadrich. "The Relationship between Precision-Recall and ROC Curves". In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 233–240. ISBN: 1595933832. DOI: 10.1145/1143844.1143874. URL: https://doi.org/10.1145/1143844.1143874.

[98] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006). ROC Analysis in Pattern Recognition, pp. 861–874. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2005.10.010. URL: https://www.sciencedirect.com/science/article/pii/S016786550500303X.

[99] Andrew Howard et al. "Searching for MobileNetV3". In: *CoRR* abs/1905.02244 (2019). arXiv: 1905.02244. URL: http://arxiv.org/abs/1905.02244.

[100] Alexander Kolesnikov et al. "Large Scale Learning of General Visual Representations for Transfer". In: *CoRR* abs/1912.11370 (2019). arXiv: 1912.11370. URL: http://arxiv.org/abs/1912.11370.

[101] Jim Winkens et al. "Contrastive Training for Improved Out-of-Distribution Detection". In: *CoRR* abs/2007.05566 (2020). arXiv: 2007.05566. URL: https://arxiv.org/abs/2007.05566.

[102] Ido Galil, Mohammed Dabbah, and Ran El-Yaniv. "A framework for benchmarking class-out-of-distribution detection and its application to ImageNet". In: (2023). DOI: 10.48550/ARXIV.2302.11893. URL: https://arxiv.org/abs/2302.11893.

[103] Paul Bergmann et al. "MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 9584–9592. DOI: 10.1109/CVPR.2019.00982.

[104] Dan Feldman. "Introduction to Core-sets: an Updated Survey". In: *CoRR* abs/2011.09384 (2020). arXiv: 2011.09384. URL: https://arxiv.org/abs/2011.09384.

[105] Deborah L. Bandalos and Meggen R. Boehm-Kaufman. "Four Common Misconceptions in Exploratory Factor Analysis". In: 2008.

[106] Michael Yuhas, Daniel Jun Xian Ng, and Arvind Easwaran. "Design Methodology for Deep Out-of-Distribution Detectors in Real-Time Cyber-Physical Systems". In: *2022 IEEE 28th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. 2022, pp. 180–185. DOI: 10.1109/RTCSA55878.2022.00025.

[107] Julián Caba et al. "Low-Power Hyperspectral Anomaly Detector Implementation in Cost-Optimized FPGA Devices". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15 (2022), pp. 2379–2393. DOI: 10.1109/JSTARS.2022.3157740.

[108] Sumaiya Tabassum Nimi et al. "EARLIN: Early Out-of-Distribution Detection for Resource-Efficient Collaborative Inference". In: *Machine Learning and Knowledge Discovery in Databases. Research Track*. Ed. by Nuria Oliver et al. Cham: Springer International Publishing, 2021, pp. 635–651. ISBN: 978-3-030-86486-6.

[109] Yiping Kang et al. "Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge". In: *SIGARCH Comput. Archit. News* 45.1 (Apr. 2017), pp. 615–629. ISSN: 0163-5964. DOI: 10.1145/3093337.3037698. URL: https://doi.org/10.1145/3093337.3037698.

[110] Paul Bodesheim et al. "Local Novelty Detection in Multi-class Recognition Problems". In: *2015 IEEE Winter Conference on Applications of Computer Vision*. 2015, pp. 813–820. DOI: 10.1109/WACV.2015.113.

[111] David Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60 (Nov. 2004), pp. 91–. DOI: 10.1023/B:VISI.0000029664.99615.94.

[112] P. Fischer, A. Dosovitskiy, and T. Brox. *Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT*. Tech. rep. 1405.5769. arXiv, May 2014. URL: http://lmb.informatik.uni-freiburg.de/Publications/2014/FDB14.

[113] Suet-Peng Yong, Jeremiah D. Deng, and Martin K. Purvis. "Modelling semantic context for novelty detection in wildlife scenes". In: *2010 IEEE International Conference on Multimedia and Expo*. 2010, pp. 1254–1259. DOI: 10.1109/ICME.2010.5583899.

[114] Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. "Textural Features for Image Classification". In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-3.6 (1973), pp. 610–621. DOI: 10.1109/TSMC.1973.4309314.

[115] Yin Cui et al. "Measuring Dataset Granularity". In: *CoRR* abs/1912.10154 (2019). arXiv: 1912.10154. URL: http://arxiv.org/abs/1912.10154.

[116] Prannay Khosla et al. "Supervised Contrastive Learning". In: *CoRR* abs/2004.11362 (2020). arXiv: 2004.11362. URL: https://arxiv.org/abs/2004.11362.

[117] Sunil Thulasidasan et al. *On Mixup Training: Improved Calibration and Predictive Uncertainty for Deep Neural Networks*. 2019. DOI: 10.48550/ARXIV.1905.11001. URL: https://arxiv.org/abs/1905.11001.

[118] Sangdoo Yun et al. "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features". In: *CoRR* abs/1905.04899 (2019). arXiv: 1905.04899. URL: http://arxiv.org/abs/1905.04899.

[119] Terrance Devries and Graham W. Taylor. "Improved Regularization of Convolutional Neural Networks with Cutout". In: *CoRR* abs/1708.04552 (2017). arXiv: 1708.04552. URL: http://arxiv.org/abs/1708.04552.

[120] Dan Hendrycks et al. *AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty*. 2019. DOI: 10.48550/ARXIV.1912.02781. URL: https://arxiv.org/abs/1912.02781.

[121] Hongxin Wei et al. *Mitigating Neural Network Overconfidence with Logit Normalization*. 2022. DOI: 10.48550/ARXIV.2205.09310. URL: https://arxiv.org/abs/2205.09310.

[122] Xuefeng Du et al. "VOS: Learning What You Don't Know by Virtual Outlier Synthesis". In: *CoRR* abs/2202.01197 (2022). arXiv: 2202.01197. URL: https://arxiv.org/abs/2202.01197.

[123] Victor Besnier et al. "Triggering Failures: Out-Of-Distribution detection by learning from local adversarial attacks in Semantic Segmentation". In: *CoRR* abs/2108.01634 (2021). arXiv: 2108.01634. URL: https://arxiv.org/abs/2108.01634.

[124] Silvio Galesso, Max Argus, and Thomas Brox. *Far Away in the Deep Space: Nearest-Neighbor-Based Dense Out-of-Distribution Detection*. 2022. DOI: 10.48550/ARXIV.2211.06660. URL: https://arxiv.org/abs/2211.06660.

[125] Antonio Torralba, Rob Fergus, and William T. Freeman. "80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.11 (2008), pp. 1958–1970. DOI: 10.1109/TPAMI.2008.128.

[126] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009.

[127] Gregory Cohen et al. "EMNIST: an extension of MNIST to handwritten letters". In: *CoRR* abs/1702.05373 (2017). arXiv: 1702.05373. URL: http://arxiv.org/abs/1702.05373.

[128] Sebastian Houben et al. "Detection of traffic signs in real-world images: The German traffic sign detection benchmark". In: *The 2013 International Joint Conference on Neural Networks (IJCNN)*. 2013, pp. 1–8. DOI: 10.1109/IJCNN.2013.6706807.

[129] Pingmei Xu et al. "TurkerGaze: Crowdsourcing Saliency with Webcam based Eye Tracking". In: *CoRR* abs/1504.06755 (2015). arXiv: 1504.06755. URL: http://arxiv.org/abs/1504.06755.

[130] Fisher Yu et al. "LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop". In: *CoRR* abs/1506.03365 (2015). arXiv: 1506.03365. URL: http://arxiv.org/abs/1506.03365.

[131] Ira Kemelmacher-Shlizerman et al. "The MegaFace Benchmark: 1 Million Faces for Recognition at Scale". In: *CoRR* abs/1512.00596 (2015). arXiv: 1512.00596. URL: http://arxiv.org/abs/1512.00596.

[132] Yandong Guo et al. “MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition”. In: *CoRR* abs/1607.08221 (2016). arXiv: 1607.08221. URL: http://arxiv.org/abs/1607.08221.

[133] Yaroslav Bulatov. “Notmnist dataset”. In: *Google (Books/OCR), Tech. Rep.[Online]. Available: http://yaroslavvb. blogspot. it/2011/09/notmnist-dataset. html* 2 (2011).

[134] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. “Human-level concept learning through probabilistic program induction”. In: *Science* 350.6266 (2015), pp. 1332–1338. DOI: 10.1126/science.aab3050. eprint: https://www.science.org/doi/pdf/10.1126/science.aab3050. URL: https://www.science.org/doi/abs/10.1126/science.aab3050.

[135] Jianxiong Xiao et al. “SUN database: Large-scale scene recognition from abbey to zoo”. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010, pp. 3485–3492. DOI: 10.1109/CVPR.2010.5539970.

[136] Yuval Netzer et al. “Reading Digits in Natural Images with Unsupervised Feature Learning”. In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. 2011. URL: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

[137] Ya Le and Xuan S. Yang. “Tiny ImageNet Visual Recognition Challenge”. In: 2015.

[138] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[139] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. “Densely Connected Convolutional Networks”. In: *CoRR* abs/1608.06993 (2016). arXiv: 1608.06993. URL: http://arxiv.org/abs/1608.06993.

[140] Mingxing Tan and Quoc V. Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In: *CoRR* abs/1905.11946 (2019). arXiv: 1905.11946. URL: http://arxiv.org/abs/1905.11946.

[141] Y. Lecun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[142] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[143] Sergey Zagoruyko and Nikos Komodakis. “Wide Residual Networks”. In: *CoRR* abs/1605.07146 (2016). arXiv: 1605.07146. URL: http://arxiv.org/abs/1605.07146.

Appendix A

Appendix

A.1 Experimental Setups Used for OOD Detection in Literature

The experimental setups- ID and OOD datasets and the model architectures on which existing OOD detection methods from literature are evaluated are shown in Table A.1.

A.2 Lengths of Feature Embeddings Extracted from Various Major Layers of EfficientNetV2-M

Table A.2 shows the 9 layers of EfficientNetV2-M [70] and the lengths of the feature embeddings extracted from them.

Layer in EfficientNetV2-M	**Feature embedding's length**
Layer 8	1280
Layer 7	512
Layer 6	304
Layer 5	176
Layer 4	160
Layer 3	80
Layer 2	48
Layer 1	24
Layer 0	24

TABLE A.2: Lengths of feature embeddings extracted from various major layers of EfficientNetV2-M

Method	Published	Datasets	Model
OpenMax [35]	Nov-15	ID: ImageNet; OOD: Fooling images by [10] , Images from the 360 classes of ImageNet-2010 that are not included in ImageNet-2012	BVLC AlexNet pre-trained on ImageNet
(Baseline) MSP [1]	Oct-16	Setup-1 - ID: CIFAR-10 and CIFAR-100; OOD: SUN, Gaussian noise images Setup 2- ID: MNIST ; OOD: Omniglot, notMNIST, CIFAR-10BW (gray-scaled CIFAR-10) , Gaussian and uniform noise images	For CIFAR-10 and CIFAR-100: 40-4 wide residual network For MNIST:MLP with 3 layers, 256 neuron-wide FC network
ODIN [3]	Jun-17	ID: CIFAR-10 and CIFAR-100; OOD: Tiny ImageNet, LSUN, Gaussian and uniform noise images Misc.:iSUN for fine-tuning hyperparameters	DenseNet (Depth L=100, growth rate k = 12) and WideResNet (WRN-28-10) both pre-trained on CIFAR-10 and CIFAR-100
Generative OpenMax [55]	Jul-17	Setup 1- ID: MNIST (0-5 digit classes); OOD: Remaining 4 classes of MNIST Setup 2- ID: HASYv2 (60 classes having 500+ images), OOD: Remaining 35 classes of HASYv2	Uses GAN and Classifier (Custom architectures of both classifier and generator/discriminator are shown in the paper)
Deep k Nearest Neighbours [43]	Mar-18	Setup 1- ID: MNIST ; OOD: NotMNIST Setup 2- ID: SVHN , OOD: CIFAR-10 Setup 3- ID: GTSRB Additionally for setup 1 and 2:OOD - ID + geometric transformations All three setups - OOD: ID images + adversarial attacks	Custom CNN with 3 convolutional layers and 2 fully-connected (FC) layers
CROSR [45]	Dec-18	ID: CIFAR-10, CIFAR-100, SVHN, Tiny ImageNet; OOD: ImageNet, LSUN. Additional experiment:Each ID dataset is split equally to make ID and OOD datasets	For MNIST:7 Layer CNN network with 5 convolutional layers and 2 FC layers. Others:A 13 layers CNN; and DenseNet (Depth L = 100, growth rate k = 24)
Outlier Exposure [39]	Dec-18	Setup 1- ID: SVHN, CIFAR-10 and CIFAR-100 ; OOD: 80 Million tiny images Setup 2- Tiny ImageNet, Places365 ; OOD: ImageNet-21k	For Places365: ResNet-18 For other datasets: WideResNet both pre-trained on ImageNet
C2AE [51]	Apr-19	ID and OOD datasets made by splitting each dataset Datasets used:MNIST, SVHN, CIFAR-10 and Tiny ImageNet Additionally - OOD: OOD dataset made from CIFAR-100 (for CIFAR-10 ID)	U-Net inspired custom network with an auto-encoder, a classifier and a decoder.
OLTR [15]	Apr-19	Setup 1- ID: ImageNet-LT; OOD: Non Overlapping classes from ImageNet-2010 Setup 2- ID: Places-LT ; OOD: PlacesExtra69 Setup 3- ID: MS1MArcFace-LT ; OOD: MegaFace Benchmark	Setup 1- ResNet-10 trained from scratch on ImageNet-LT Setup 2- ResNet-152 pre-trained on ImageNet fine-tuned for PlacesLT Pretrained ResNet-50 (for MS1MT) ImageNet-LT; Places-LT
Generalised ODIN [37]	Mar-20	ID: SVHN, CIFAR-10, and CIFAR-100 OOD: Tiny ImageNet(crop), Tiny ImageNet(resize), LSUN-Crop, LSUN-Resize, iSUN, Uniform random images, and Gaussian noise image	DenseNet (Depth L=100, growth rate k = 12), WideResNet (WRN-28-10) (did not specify pre-trained or not)
Tempered Mix-up [62]	Sep-20	Setup 1- ID: CIFAR-10; OOD: Tiny ImageNet (178 non-overlapping classes) Setup 2- ID: MNIST; OOD: EMNIST-Letters Setup 3- ID and OOD: Same as OpenMax ([35])	Setup 1- ResNet-32 pre-trained on CIFAR-10 Setup 2- LeNet++ trained on MNIST Setup 3- ResNet-18 pre-trained on ImageNet
Energy-based OOD detection [2]	Apr-21	IDs:SVHN, CIFAR-10, and CIFAR-100 ; OOD: Textures, SVHN, Places365, LSUN-Crop , LSUN-Resize, and iSUN Misc.:Outlier dataset:Tiny ImageNet disjoint from CIFAR-10 and CIFAR-100	WideResNet pre-trained on CIFAR-10 and trained on each ID dataset separately
MOS [47]	May-21	ID: ImageNet; OOD: iNaturalist (110 classes not in ImageNet), SUN and Places365 (50 classes from each that are non-overlapping with ImageNet), DTD: All classes	Google BiT-S models as feature extractor in all experiments. The models are trained on ImageNet-1k, with ResNetv2 architectures (specifically BiT-S-R101x1 for main results)
GradNorm [4]	Oct-21	same as MOS [47]	same as MOS [47]
OpenGAN [61]	Oct-21	Setup 1- ID: MNIST 0-5 digit classes; OOD: MNIST 6-9 digit classes Setup 2- ID: 5 classes OF CIFAR-10; OOD: remaining 5 class of CIFAR-10 Setup 3- SVHN (ID: 0-5 digit classes; OOD: 6-9 digit classes Setup 4- ID: Randomly selected 20 classes from TinyImageNet, OOD: Remaining 180 classes of Tiny ImageNet	ResNet-18 trained on ID for each setup as the discriminator in GAN (Note:There are more setups aiming at segmentation and pixel-level OSR and are not mentioned here)
ReAct [40]	Nov-21	ID: ImageNet; OOD: iNaturalist (110 classes not in ImageNet), DTD (Textures) SUN and Places365 (50 classes non-overlapping classes with ImageNet) .	ResNet-50 pre-trained on ImageNet
CIDER [50]	Mar-22	Same as MOS [47]	ResNet-18 for CIFAR-10, ResNet-32 for CIFAR-100
PCA FRE-based [5] OOD detection	Mar-22	Setup 1- ID: CIFAR-10, CIFAR-100, OOD: SVHN, LSUN Setup 2- ID: SVHN, OOD: CIFAR-10, LSUN Setup 3- (Anomaly detection) MVTec	ResNet-18 for CIFAR-10, Wide-ResNet for CIFAR-100, ResNet-20 for SVHN EfficientNet-B5 for Anomaly detection
POEM [53]	Jun-22	ID: CIFAR-10 (setup 1), CIFAR-100 (setup 2) OOD (for both setups): SVHN, DTD, Places365, LSUN-crop, LSUN-resize, and iSUN Misc.:Down sampled version of ImageNet (ImageNet-RC) as auxiliary outlier dataset	DenseNet-101
Distance to the k-nearest neighbours DkNN [6]	Jun-22	Setup 1- ID: CIFAR-10, OOD: DTD, SVHN, Places365, LSUN-C, and iSUN Setup 2- Same as MOS	Setup 1- ResNet-18 trained on CIFAR-10 Setup 2- ResNet-50 trained on ImageNet

TABLE A.1: An overview of the datasets and model architectures used in different DNN-based methods (sorted chronologically). Note: Unless specified, the ImageNet dataset refers to the ImageNet-1K from ILSVRC2012.

References Datasets:80 Million tiny images [125], CIFAR10 and CIFAR100 [126], DTD [77], EMNIST [127], GTSRB [128], HASYv2 [58], ImageNet [8], iNaturalist [74], iSUN [129], LSUN [130], MegaFace Benchmark [131], MNIST [57], MS1MArcFace [132], MVTec [103], NotMnist [133], Omniglot [134], Places [76], SUN [135], SVHN [136], TinyImageNet [137]
Network architectures:AlexNet [138], DenseNet [139], EfficientNet-B5[140], LeNet [141], ResNet [142], WideResNet [143],

A.3 Variants of the Baseline Method

In this section, the variants of the Baseline MSP method are described and compared.

A.3.1 Softmax Probability Margin

This OOD detection method is an extension of the first baseline MSP. Instead of the MSP of an image, the margin (difference) between the highest softmax probability and the second highest softmax probability is taken as the OOD score. This baseline is called softmax probability margin (SPM). The underlying assumption for this method is that a classifier tends to be uncertain about the class to which an out-of-distribution image belongs, making the softmax probabilities nearly identical for all classes. This makes the margin between the highest and the second-highest softmax probabilities to be low. Whereas, the opposite can be assumed to be true for an in-distribution image, and hence the margin would be higher. softmax probability margin (SPM) for an input image x is given by:

$$SMP(x) = S_{sorted_C}(x) - S_{sorted_{C-1}}(x) \ , \tag{A.1}$$

where S_{sorted} is the set of softmax probabilities sorted in ascending order given by the classifier trained on a total of C classes. $S_{sorted_C}(x)$ and $S_{sorted_{C-1}}(x)$ are the highest softmax probability and the second-highest softmax probability for an input image x. Similar to the MSP, the SPM score can be compared against a predetermined threshold to classify an input as ID or OOD.

A.3.2 Results

Table A.3 shows the comparison of the baseline methods when applied on logits and when applied on softmax probabilities for an OOD detection task. As mentioned in [64], the Baseline MSP method when applied on the logits gives better OOD performance.

Method	**ID: Pap.50; OOD: Places365 (TPR@FPR1)** ↑	**ID: CUB80; OOD: Places365 (TPR@FPR1)** ↑	**ID: iNat.2100; OOD: Places365 (TPR@FPR1)** ↑
Maximum softmax probability (MSP)	**22.99**	90.83	51.75
Softmax probability margin (SPM)	21.39	89.74	50.93
Maximum of output logits	22.46	**99.13**	**54.33**
Output logits margin	13.37	88.21	49.68

TABLE A.3: Comparison of OOD detection performance between the variants of baseline methods

A.4 Additional Density-based Methods

Two more density-based methods were evaluated but are not presented in the main results.

A.4.1 Local Outlier Factor

The Local outlier factor (LOF) method is a popular density-based method introduced in [31] for detecting the existing outliers in a dataset. The core concept of the LOF method is combining the DkNN method along with a so-called 'reachability distance' between the data points. LOF when adapted for OOD detection consists of the following steps :

- Reachability distance: The reachability distance is the maximum distance between two points, where one point is considered to be reachable from the other if it is closer than a given radius. In other words, the reachability distance between two points is a measure of how easily one point can be reached from another.

- Local reachability density: The local reachability density (LRD) is the key concept in the LOF algorithm. It is a measure of how dense the neighbourhood of a data point is, with respect to its k-nearest neighbours. A point with a high LRD implies that it has a dense neighbourhood. The LRD of a point is defined as the inverse of the average reachability distance of its k-nearest neighbours. For a test image x, let f_x be the normalised feature embedding. If the set of k nearest neighbour feature embeddings (from the training set) of f_x is given by N_{x_k}, then the LRD of f_x is calculated as:

$$lrd_k(f_x) = \frac{1}{\frac{\sum_{f_i \in N_{x_k}} reach\_dist_k(f_x, f_i)}{k}} \ , \tag{A.2}$$

  where $reach\_dist()$ is the reachability distance and is calculated as :

$$reach\_dist_k(f_x, f_i) = maximum\left\{kdist(f_i, k), dist(f_x, f_i)\right\} \ , \tag{A.3}$$

  where $kdist(f_i, k)$ is the distance between f_i and its k^{th} nearest neighbour and $dist(f_x, f_i)$ is the distance between f_x and f_i. In this work, the $dist(f_x, f_i)$ function returns the Euclidean distance between f_x and f_i.

- Local outlier factor: LOF measures the degree of 'outlierness' of a data point by comparing the local density of the point with the local densities of its k-nearest neighbours. Using the LRD Equation A.2, the LOF for the feature embedding f_x can be calculated as:

$$LOF_k(f_x) = \frac{\sum_{f_i \in N_{x_k}} \frac{lrd_k(f_i)}{lrd_k(f_x)}}{k} \ . \tag{A.4}$$

In a way, LOF is the ratio of the average reachability distance of its k-nearest neighbours (kNNs) to its reachability distance. The intuition behind this definition is that a point with a high LOF score is an outlier if its kNNs are much denser than it is. Conversely, a point with a low LOF score is not an outlier if its kNNs are similarly dense. To classify test points as ID or OOD, a threshold is chosen such that points with a LOF score above the threshold are classified as outliers. The choice of threshold is based on the distribution of the LOF scores for the validation set of the in-distribution data.

A.4.2 Local Distance-based Outlier Factor

The authors in [32] introduce the local distance-based outlier factor as a better alternative to the Local outlier factor for the task of outlier detection, especially when the data points are scattered.

LDOF uses the relative position of a test data point to its neighbours in the training set to measure the degree to which the data point deviates from its neighbourhood [32]. The higher the LDOF value for a data point, the more likely that it is an outlier. In this work, LDOF-based OOD is implemented in the following steps:

- **Indexing the normalizing train feature embeddings** The normalised feature embeddings are indexed and added to the kNN search space. Elaborate step-by-step details are given in steps 1-3 in the kNN section (Subsection 3.2.6.2)

- **kNN distance of a test image's feature embedding** The kNN search is performed on the feature embedding f_x of the test image x. Let N_{x_k} be the set of the k-nearest neighbours in the training set for the feature embedding f_x. Then the distances from f_x to all the feature embeddings in the set N_{x_k} are averaged. This can be mathematically represented by the following equation:

$$\overline{d}_{f_x} = \frac{1}{k} \sum_{f_i \in N_{x_k}} dist(f_i, f_x) \ , \tag{A.5}$$

  where k is the number of neighbours, N_{x_k} is the set of these neighbouring feature embeddings. $dist()$ function is a function that calculates a distance metric between two points. In this work, the distance metric is chosen to be Euclidean distance.

- **Calculation of inner distance** A so-called 'inner distance' is calculated for f_x. This is the average distance among the set of neighbours in N_{x_k}. This is given by:

$$\overline{D}_{f_x} = \frac{1}{k * (k-1)} \sum_{f_i, f_{i'} \in N_{x_k}, i \neq i'} dist(f_i, f_{i'}) \ . \tag{A.6}$$

- **Calculation of LDOF** The local distance-based outlier factor for feature embedding f_x is given by :

$$LDOF_k(f_x) = \frac{\overline{d}_{f_x}}{\overline{D}_{f_x}} \ . \tag{A.7}$$

The LDOF is directly used as the OOD score in the OOD decision Equation 2.1.

A.4.3 Results

The results for near-OOD detection are shown in Table A.4. Overall, the LOF method outperforms the other two methods. However, the SLOF method is computationally less expensive (compared to LOF) and achieves similar OOD detection performance. This was also the main motivation to show the SLOF method in the main results.

	ID: Pap.50; OOD: Pap.62			**ID: CUB80; OOD: CUB120**			**ID: iNat.2100; OOD: iNat.2989**		
Method	AUROC ↑	TPR@FPR1 ↑	FPR@TPR99 ↓	AUROC ↑	TPR@FPR1 ↑	FPR@TPR99 ↓	AUROC ↑	TPR@FPR1 ↑	FPR@TPR99 ↓
LOF	**86.01**	**39.57**	**65.78**	**85.85**	20.74	**79.86**	**75.4**	**10.77**	**94.18**
SLOF	85.34	10.96	68.81	85.02	**22.93**	85.52	74.77	10.3	94.25
LDOF	79.71	9.36	69.02	79.56	20.96	87.97	53.69	3.06	96.83

TABLE A.4: Comparison of OOD detection performance of the density-based methods on the near-OOD detection task

A.5 Results - Variants of DkNN Method

As mentioned in Subsection 3.2.6.2, instead of the distance to the k^{th} nearest neighbour, a few other statistical methods are also considered as OOD scores. Table A.5 shows these results.

	ID: Pap.50; OOD: Pap.62			**ID: CUB80; OOD: CUB120**			**ID: iNat.2100; OOD: iNat.2989**		
OOD Score calculated using	AUROC ↑	TPR@FPR1 ↑	FPR@TPR99 ↓	AUROC ↑	TPR@FPR1 ↑	FPR@TPR99 ↓	AUROC ↑	TPR@FPR1 ↑	FPR@TPR99 ↓
Distance to the k^{th} NN (Regular DkNN)	86.56	56.42	69.81	86.60	23.14	**75.71**	80.59	17.85	96.10
Mean of distance to NN and k^{th} NN	**90.21**	58.56	64.66	**86.66**	**24.02**	78.06	80.59	**18.66**	**96.00**
Mean of distances to k NNs	87.23	60.70	64.66	86.62	**24.02**	78.14	80.59	17.85	**96.00**
Mode of distances to k NNs	86.21	6.42	69.73	86.43	22.49	77.19	80.59	17.88	96.09
Median of distances to k NNs	86.72	50.27	64.66	86.60	**24.02**	**75.71**	80.59	17.85	96.10
Standard deviation of distances to k NNs	87.06	**60.96**	**64.26**	86.62	**24.02**	78.07	80.59	17.85	96.01

TABLE A.5: Near-OOD detection results for the variants of the DkNN method. NN is short for nearest neighbour.

A.6 Domain Similarity and Model Architecture Dependency

Figures A.1 and A.2 show the domain similarity scores calculated using different architectures for CUB80 and iNaturalist2100 datasets respectively.

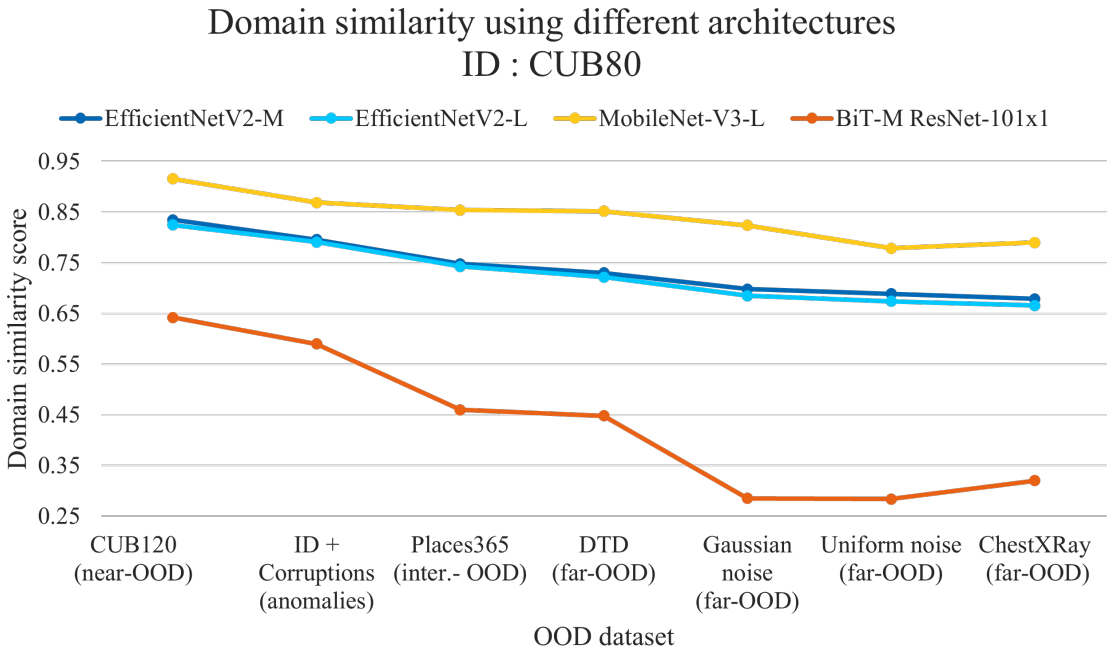


FIGURE A.1: Domain similarity scores using different model architectures for CUB80 dataset

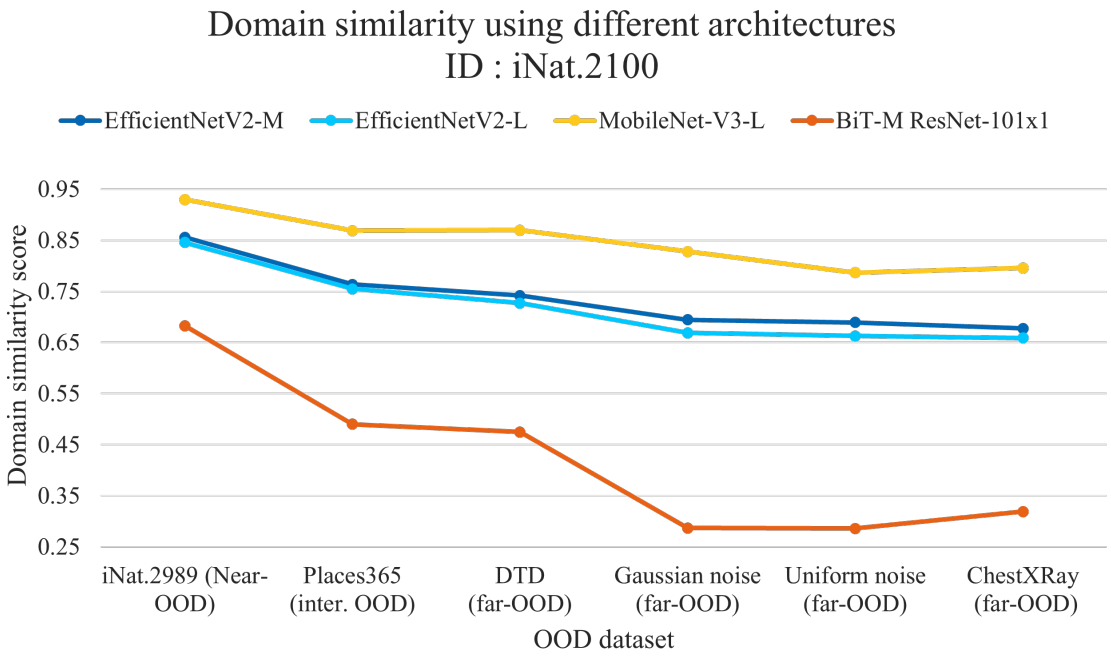


FIGURE A.2: Domain similarity scores using different model architectures for iNaturalist2100 dataset