Out-of-Distribution Detection using Inter-level Features of Deep Neural Networks

by

Jamil Fayyad

B.Sc., American University of Sharjah, 2009
M.Sc., American University of Sharjah, 2012

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE COLLEGE OF GRADUATE STUDIES

(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

May 2023

The following individuals certify that they have read, and recommend to the College of Graduate Studies for acceptance, a dissertation entitled:

Out-of-Distribution Detection using Inter-level Features of Deep Neural Networks

submitted by Jamil Fayyad in partial fulfilment of the requirements of the degree of Doctor of Philosophy

Dr. Homayoun Najjaran, School of Engineering
Supervisor

Dr. Sumi Siddiqua, School of Engineering
Co-Supervisor

Dr. Dean Richert, School of Engineering
Supervisory Committee Member

Dr. Rudolf Seethaler, School of Engineering
Supervisory Committee Member

Dr. Liwei Wang, School of Engineering
University Examiner

Dr. William Melek, University of Waterloo
External Examiner

Abstract

The work presented in this thesis addresses the problem of Out-of-Distribution (OOD) detection in deep learning-based classifiers. The emphasis is to investigate real-world settings, where both the In-distribution (ID) and OOD samples are hard to distinguish due to their high semantic similarities. Despite the significant performance of classification networks, they fail to correctly handle OOD samples. During training, the closed-world assumption is heavily relied upon. It is assumed that models are only forwarded samples that share similar distributions to the training data. In dynamic environments, the assumption is strict and leads to severe degradation in performance. While uncertainty quantification methods are elegant solutions to represent predictive confidence, the presented analysis of said methods illustrates multiple challenges in OOD detection. Motivated by the shortcomings of uncertainty-based detectors, a novel mechanism is proposed to efficiently detect OOD samples. The mechanism leverages features extracted from intermediate layers of a classifier and examines their activations. These activations are distinctive for ID samples and can be utilized to distinguish dissimilar data, even if they are classified to the same label. To verify the performance, the proposed method was first amalgamated with an uncertainty-based classifier and tested to detect OOD samples selected

to be similar to the ID samples. The method significantly outperforms the uncertainty threshold-based method for OOD detection. Furthermore, the approach was implemented with common classifiers and has shown improved performance on different common OOD datasets.

Lay Summary

Deep learning algorithms have become widely adopted in various applications, such as image classification and labeling, due to their remarkable ability to learn patterns from extensive training datasets and surpass human performance in accurately identifying different objects. However, a significant drawback of deep neural networks is their inability to provide accurate and meaningful outputs when faced with inputs that lie outside the training data domain. This limitation poses potential risks and challenges, particularly in safety-critical applications, which hinder the deployment of these networks. The presented work aims to address this limitation by enabling the networks to identify unknown samples instead of generating false predictions. By improving the models, the outcomes facilitate the safe deployment of networks in real-world applications, harnessing their full performance potential while ensuring robustness.

Preface

The work presented in this thesis was conducted at the Advanced Control and Intelligent Systems (ACIS) Laboratory at the University of British Columbia, Okanagan under the supervision of Dr. Homayoun Najjaran. The work includes a comprehensive literature review, determining the research gap, formalizing the concepts, and experimental validations. A summary of publications and main thesis components are listed:

- Chapter 1 identifies the research gap and highlights the motivations and benefits of the proposed concept.

- Chapter 2 presents an extensive literature review of the presented topic. Part of the chapter is published in [1]: J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran, "Deep learning sensor fusion for autonomous vehicle perception and localization: A review," Sensors, vol. 20, no. 15, p. 4220, 2020.

- Chapter 3 provides an in-depth analysis of uncertainty-based OOD detection. Additionally, the chapter delves into the significance of appropriately selecting the OOD dataset to accurately assess the algorithm's performance. Results from Chapter 3 are submitted for review in: J. Fayyad, K. Gupta, D. Gruyer, and H. Najjaran, "Analysing Uncer-

tainty Quantification for Out-of-Distribution Detection in Complex, Real-World Datasets" Machine Vision and Applications.

– The main concept of Dominant Inter-level Feature Exploitation (DIFE) is presented in Chapter 4. Results are submitted for review in: J. Fayyad, K. Gupta, N. Mahdian, D. Gruyer, and H. Najjaran, "Exploiting Classifier Inter-level Features for Efficient Out-of-Distribution Detection" Image and Vision Computing.

A fruitful collaboration with fellow colleagues at the ACIS lab has led to a number of publications in the domain of computer vision, perception, and the optimization of deep networks:

– Developing an unsupervised multi-object tracking algorithm based on deep neural network: N. Mahdian, A. Shojaeinasab, J. Fayyad, and H. Najjaran, "Unsupervised multi-object tracking algorithm with adaptive track-matching," 2023, submitted to IROS 2023.

– The quantization of deep neural networks using deep reinforcement learning and KL divergence metric, to optimize the inference latency and memory footprint: Y. Al-Younes, J. Fayyad, and H. Najjaran, "Systematic mixed-precision post-training quantization," 2023, under review, Neurocomputing.

– Improving 3D map generation and label generation in dynamic environments [2]: J. Roch, J. Fayyad, and H. Najjaran, "Dopeslam: High-precision ros-based semantic 3d slam in a dynamic environment," Sensors, vol. 23, no. 9, p. 4364, 2023

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AE Autoencoders. 11

AI Artificial Intelligence. 1

AUPR Area Under the Precision-Recall curve. 71

AUROC Area Under the Receiver Operating Characteristic curve. 71

BNN Bayesian Neural Networks. 24

CNN Convolutional Neural Network. 6

DBN Deep Belief Networks. 11

DIFE Dominant Inter-level Feature Exploitation. vii

DL Deep Learning. 2

EDL Evidential Deep Learning. 8

FN False Negative. 15

FP False Positive. 15

ID In-Distribution. 3

ILSVRC ImageNet Large Scale Visual Recognition Challenge. 13

MCD Monte Carlo Dropout. 25

MCMC Markov Chain Monte Carlo. 25

MLP Multi-Layer Perceptron. 12

ND Novelty Detection. 38

OOD Out-of-Distribution. 3

OSR Open Set Recognition. 38

RNN Recurrent Neural Networks. 11

SL Subjective Logic. 43

TN True Negative. 15

TP True Positive. 15

VGG Visual Geometry Group. 14

Acknowledgements

I want to start by expressing my heartful gratitude and deepest appreciation to my father Walid and my mother Amira for their continuous love and unending support. The person I have evolved into today is a direct reflection of your selfless sacrifices. Thank you for believing in me. I would like to extend my deepest gratitude to my wife, Zaheia, who has been by my side throughout this entire journey. Your persistent support, motivation, meticulous proofreading, and ability to keep me sane have been absolutely indispensable in helping me reach this significant milestone. I am eternally grateful for my son, Waleed. A significant portion of your early growing years overlapped with my academic journey. I promise to make up for the time that has passed by cherishing every moment to come. I finally would like to thank my sisters, Yara, Muneera, and Noor. Thank you for always lending me your ears and being there for me when I needed support.

I am deeply appreciative of my advisor, Dr. Homayoun Najjaran, for his continuous encouragement and guidance. Today, I have understood the significance behind the often-used phrase found in acknowledgments, *"Words can't express my gratitude towards my advisor"*. In addition to his exceptional knowledge, Dr. Najjaran exemplifies true kindness and possesses the ability to invest in his students and keep them motivated. I am sincerely

grateful for his unwavering support through both the successes and challenges I have encountered. I want to also extend my appreciation to Dr. Dominique Gruyer for his guidance and technical support. I would also like to also thank my committee members, Dr. Dean Richert and Dr. Rudolf Seethaler for their insightful comments and feedback.

Last, but not least, I would like to express my appreciation to all my colleagues at the Advanced Control and Intelligent Systems (ACIS) lab. Our technical discussions, casual conversations, shared lunch breaks, outdoor hikes, and various activities have brought me immense joy and fulfillment. I am particularly grateful to Dr. Kashish Gupta and Navid Mahdian for their contribution to my work.

Dedication

To my beloved parents . . .

To my lovely wife, wonderful son, and soon-to-come daughter . . .

To my dearest grandparents . . .

To my homeland . . .

Chapter 1

Introduction

1.1 Motivation

***Q**: What is Artificial Intelligence?*

***A**: Artificial intelligence (AI) refers to the ability of a machine or computer system to perform tasks that would normally require human intelligence, such as learning, problem-solving, and decision-making.*

***Q**: Does it have any limitations?*

***A**: AI systems have limitations in the sense that they can only perform tasks and make decisions within the scope of their programming.*

Can you believe that the conversation above was generated by a chatbot trained to replicate human conversational behavior? In this time and age, AI has practically replaced a lot of tasks that previously needed to be executed by human beings only. The evolution of AI has been so rapid over the past decade that keeping up with its latest trends has been quite a challenge. With the rapid evolution of AI, it is crucial that the newly developed systems are applicable to real-world scenarios away from the simulated laboratory settings. Hence, AI requires to be transferable to real-life situations to account for naturalistic environmental shifts. For example, Meta's

language model, *Galactica*, was immediately shut down after receiving criticism for generating dangerous and misleading information about science [3]. Therefore, testing models beyond what they are trained on and investigating their failure cases are prerequisites for us to understand the development of the AI system and potentially produce novel models that can successfully mimic nature and overcome such complications. The motivation of the proposed work aims to bridge the gap between the laboratory testing of AI approaches, and their real-world deployment.

There are a variety of fields where Deep Learning (DL) models have been successfully applied; however, due to their black-box nature, it is difficult to gain a comprehensive understanding of their internal behavior. The ability to explain the model's decision is essential to build trustworthy models, especially in applications with tight error tolerance. To elaborate, consider a scenario where a deep binary classification network is trained to recognize examples of two classes. Through conventional effective training strategies, the classifier would achieve a reasonable accuracy that would allow generalization to data of a similar distribution at inference. Issues arise when a sample that does not belong to the distribution of either class is forwarded to the network. In such instances, models would fail to generate interpretable results, despite their high validation accuracy. In fact, due to the design architecture, the network would be forced to classify the unknown sample into one of the class labels. This described behavior in the aforementioned example is generalizable to include the multilabel classification networks as well. To further investigate such shortcomings, it would be beneficial to look deeper and evaluate the activation of different layers of the network for dis-

tinct samples. Our illustrative experiment uses a simple AlexNet network [4] with five convolutional layers as an image classifier. The network is trained on a dataset with two classes: cars and buses. After training is completed, testing samples are divided into two categories. The first group includes images of cars and buses and is referred to as In-Distribution (ID) samples, while the second group includes random images of pedestrians, and is referred to as Out-of-Distribution (OOD) samples. The objective is to analyze how the network interacts with samples of both groups. First, an ID image (car image) is forwarded to the network. The average value of each activation filter extracted from the first layer is obtained. The same is performed with an OOD image (also classified as a car). Then, the distance between the activation vectors of both images is calculated. Finally, the process is repeated over all the other layers of the network. The results are compared for two distinct cases: The first case represents the distance between two similar ID samples, while the second case represents the distance between an ID and an OOD sample. It can be observed that intermediate layers hold a larger distance measurement for samples of the second case compared to the measurement of the first case. Both distances gradually drop by the time it reaches the last output layer. The intuition here is that the network is forced to approximate semantic information in OOD samples in order to classify them into a single label. Looking at earlier layers, the results indicate that distinct and important semantic information corresponding to the dissimilarity between the ID and OOD samples can be detected at those layers of the network. Hence, the activations of these layers can be leveraged to distinguish OOD samples. This observation motivates the work presented

in this thesis.

In the past few years, OOD detection has become an active research field in DL. In safety-critical applications, the ability to distinguish unknown data allows models to operate more reliably. A "trivial" solution would be to use OOD data, or a portion of the data, for model training or calibration. In real-life settings, OOD data is very diverse and cannot be easily defined, hence it is not accessible throughout the detection process. Leveraging the available information inherent in the ID features, on the other hand, is a potential solution to the problem, and forms another motivation for the presented work.

1.2 Problem Statement

DL is a renowned approach for a variety of domains and applications. The capacity of the method to analyze and effectively handle massive amounts of data has pushed the possibilities beyond what human beings are capable of comprehending. DL, like every revolutionary approach, has its drawbacks that need to be properly addressed. Standard networks are incapable of representing the confidence of their predictions. The problem extrapolates when deploying these black-box models in safety-critical and high-risk applications like autonomous driving, medical diagnosis, and fraud detection. In such systems, it is critical to identify situations when the model is less confident, to mitigate any possible risk.

To further understand the complication, consider a network that is trained to perform image classification. The standard Softmax function is used at

the output layer to transform raw activation vectors into class probabilities. Intuitively, the output of Softmax would be assumed to provide a perfect measure of network confidence. All the images that are related to the training dataset should correspond to a sharp distribution over Softmax outputs (indicating high confidence), while any unknown and unrelated images should result in a uniform distribution (indicating total uncertainty). Unfortunately, this is not the case with DL networks. Unfamiliar images will still be classified to one of the class labels, probably with a high confidence score too. The reason is that the outputs of DL classifiers are not well-calibrated [5]. DL networks are often overconfident, they generally assign high output scores to their predictions even if the predictions are incorrect. As a result, the output scores of DL networks are unreliable indicators of the network's confidence.

The aforementioned example can be expanded and generalized to similar limitations of other DL applications. In general, the performance of DL algorithms is constrained by the data distribution used in training and evaluation. To ensure good model performance, the distribution of the evaluation data should be similar to the distribution of the training data. A slight difference between the two distributions can cause the performance to degrade noticeably. There is a variety of sources that can cause such discrepancy, new samples that have not been seen during training are one of them. In real-world settings, it is unfeasible to ensure that all data forwarded to the trained model is drawn from a similar distribution. For a safe and reliable operation, models should assign low confidence for all OOD samples and successfully detect them.

A plethora of approaches has been devoted to studying how DL models can efficiently represent the confidence of their predictions. Recent studies have investigated uncertainty quantification techniques as a solution. A range of methods proposes adjustments to the network configuration [6] (Deep ensembles), adjustments to their weights [7] (Bayesian Inferencing), or adjustments to their likelihood and loss functions [8] (Evidential methods). Further details of uncertainty quantification methods are included in Section 2.2. In general, uncertainty quantification methods are presumed to generate low uncertainty values for all known ID data, and high uncertainty values for all unknown OODs. This assumption does not hold true, especially in real-world settings where there is a significant feature overlap between ID and OOD samples. As a result, uncertainty quantification techniques are not very reliable for the task of OOD detection.

Ultimately, the objective is to establish DL models that can accurately classify ID data and simultaneously detect OOD samples. In practical scenarios, ID and OOD samples often share similar semantic features, Therefore, it is important to ensure that the evaluation dataset contains OOD samples that are correlated with the IDs. The proper selection of the OOD dataset is crucial for bridging the gap between model performance in controlled environments and real-world deployment.

To summarize, the following points address the challenges:

1. Classical Convolutional Neural Network (CNN) classifiers are overconfident with their predictions. The use of the Softmax activation at the output layer generates confidence scores that are not well calibrated.

2. With minor adjustments, uncertainty quantification approaches enable the network to generate predictions and estimate uncertainties. Placing a fixed threshold value on the uncertainty output, however, is insufficient for OOD detection.

3. The detection of OOD samples should be done without explicitly training networks on OOD data. Moreover, a proper evaluation of OOD detection algorithms requires a careful selection of an OOD dataset. Practical applications often encounter OOD samples that are similar to the ID samples.

1.3 Objectives and Contributions

DL approaches have achieved state-of-the-art performance in a wide range of applications. Most often, however, it is falsely assumed that samples at inference follow a similar distribution as the training data. This assumption impairs models' ability to handle unseen samples during inference. This research focuses on CNN-based classifiers subjected to OOD samples. The aim is to enhance the robustness of existing classification networks by improving their OOD detection. The following are the objectives of the work presented:

– Analyze the efficacy of uncertainty quantification approaches to detect OOD samples. The detection approach is based on a fixed threshold value applied to the quantified uncertainty.

– Formalize a general framework that exploits dominant features extracted from intermediate layers to detect OOD samples. The frame-

work learns from ID samples only and does not require access to OOD data.

– Integrate the proposed framework with uncertainty-aware classifiers to enhance OOD detection and replace the dependence on the uncertainty threshold.

The introduction of OOD detection in DL-based classification networks has been demonstrated to be a vital step for practical applications. Classifiers must indicate when they *"don't know"* how to classify a sample rather than assigning it one arbitrary label. In fact, the benchmarking of classifiers' performance should include OOD detection metrics, rather than relying on accuracy alone. Over the last five years, researchers have dedicated great effort to developing efficient OOD detection techniques. It is, however, challenging to select the proper evaluation dataset for OOD detection algorithms. The contributions of this thesis are highlighted below:

– In-depth analysis of evidential-based uncertainty quantification algorithms, including a thorough evaluation of OOD detection, using a real-world dataset with relevant OOD samples (Chapter 3, Section 3.1, and 3.2)

– A novel concept to detect semantic shift OOD samples based on extracted dominant features of ID samples, and without training on OOD samples. The concept is referred to as DIFE (Chapter 4, Section 4.1)

– Integrate DIFE with Evidential Deep Learning (EDL) uncertainty quantification and evaluate OOD detection performance on a range of experiments with different datasets (Chapter 4, Section 4.2)

– Propose a distance-based OOD detection framework that is motivated by the inter-level dominant features of ID samples. (Chapter 4, Section 4.3)

Chapter 2

Background

2.1 Deep Learning Image Classification

DL is a subset of artificial intelligence and machine learning that tries to mimic the functions of the human brain. The algorithms involve creating manifold networks that have multiple layers, allowing it to process raw data and extract certain patterns to perform complex and intelligent tasks. The core concept of DL is based on Artificial Neural Networks (ANN), which can be traced back to 1943 when Walter Pitts and Warren McCulloch [9] took the first steps towards building a model that is based on the working principle of human's brain neural networks. While the basics of DL were found long ago, its recent vast emergence is due to the development of powerful computing machines and the availability of "big data" needed to train the models. Recently, DL is being extensively used in many different applications as object detection [10], environment segmentation, semantic object identification, healthcare [11], self-driving vehicles [12, 13], and finance [14], to name a few. There exist several different algorithms that are listed under the category of DL. Each technique has its unique properties and hence is used for a certain application where the goal is to achieve optimal performance. The frequently used DL methods can be listed as:

1. **Convolutional Neural Networks:**

   CNN is a feedforward network with convolution and pooling layers. The architecture is powerful in finding the relationship among image pixels and hence is used in applications related to computer vision as image detection and classification. [15, 16]

2. **Recurrent Neural Networks:**

   Recurrent Neural Networks (RNN) is a class of feedback networks that uses previous output samples to predict new data samples. RNNs are often used to process sequential data; both the input and the output are represented by a sequence of data. Applications of RNN include image captioning [17], data forecasting [18], and Natural Language Processing (NLP) [19].

3. **Deep Belief Networks:**

   Deep Belief Networks (DBN) is a multi-layer generative energy-based model with a visible input layer and multiple hidden layers. DBN assigns probabilistic values to its model parameters. It is used in application as acoustic modeling [20], and collaborative filtering [21]

4. **Autoencoders:**

   Autoencoders (AE) are a class of neural networks that tends to learn a representation of data in an unsupervised manner. AE consists of an encoder and decoder and can be trained by minimizing the construction loss between the input and the output. Applications of AE include dimensionality reduction [22], image retrieval [23], and data denoising [24].

Before CNN was introduced (2012), Multi-Layer Perceptron (MLP) was heavily used in image recognition and classification. MLP is a feedforward, fully connected neural network, and it consists of an input layer, a hidden layer, and an output layer. With current advancements, it is concluded that MLP has many limitations and cannot be a sufficient tool due to the following disadvantages. First, it has a growing number of parameters that need to be trained. Second, it loses spatial information and pixel arrangement of an image. Third, it is not translation-invariant. On the other hand, CNN is a subset of DL algorithms that uses convolution operation to process pixels in images. It has a different architecture compared to that of an MLP; the layers are organized in three dimensions: width, height, and depth. Additionally, the neurons of CNN are not fully connected to the layers. A general CNN usually consists of the following layers, as shown in Figure 8:

- **Input layer:** Input layer contains the three-dimensional data information in the form of *Width* × *Height* × *Channels*. For RGB images, *Channels=3*

- **Convolution layers:** Convolution operation is performed to extract important features from the input image ($W \times H$). A filter of size $K \times K$ slides across the image, and an elementwise multiplication is performed between the pixel values and the filter weights. The stride (S) parameter controls the amount of movement of the filter, while the padding parameter (P) adds zeros to the edge of the image to maintain a specific size. The new image dimensions after convolution are given by:

$$n_{out} = \frac{n_{in} + 2P - K}{S} + 1 \tag{2.1}$$

– **Pooling layers:** Located between convolution layers and help in minimizing the computational cost by reducing some, but maintaining the most dominant, spatial information of the convoluted image.

– **Fully connected layer:** Performs linear transformation and Connects every neuron in the current layer to the neuron in the next layer to perform classification.

– **Output Layer:** It stores the final output as classification probabilities.

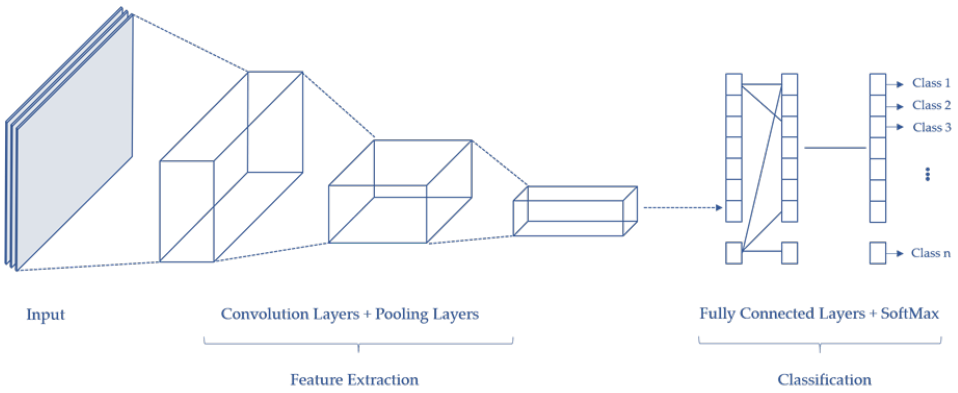


Figure 2.1: General architecture of a Convolutional Neural Network

Standard classification architectures

CNN was first introduced by Lecun et al. [25]. Since then, its present prominence stems from its use by top-ranking methods winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [26]. The rising complexity of input data has led CNN architectures to undergo significant advances throughout the years. The popular architectures used in image

classification are reviewed here:

1. **LeNet:**

   LeNet is a simple CNN network introduced by Yann LeCun in 1998 [25]. The network consists of 2 convolution layers and 3 fully connected layers and 60,000 parameters. The network was trained to classify the 10-class MNIST handwritten digits.

2. **AlexNet:**

   AlexNet was proposed by Krizhevsky et al. [4] and won the 2012 ILSVRC challenge. The proposed architecture has an error rate of 16%, which is 10% lower than the second-runner approach. AlexNet takes a 256 x 256 RGB image and has 60 million parameters. It consists of 5 convolution layers and 3 fully connected layers.

3. **VGG:**

   The Visual Geometry Group (VGG) network was founded in 2014 by a group of researchers from Oxford University [27]. The network ranked second in the 2014 ILSVRC competition with less than a 7% error rate. VGG has a deep architecture with 16 layers in VGG-16 and 19 layers in VGG-19. All layers are activated with the ReLU activation function, and the input is a 224x224 RGB image.

4. **GoogLeNet:**

   The GoogLeNet [28] was introduced in 2014 after winning first place in the 2014 ILSVRC challenge, achieving an error rate of 6.7%. The network has 22 deep layers and uses the Inception modules introduced in [28]. The input image dimensions of the network are 224x224.

5. **ResNet:**

   The Microsoft team's Residual Networks [29] took first place in the 2015 ILSVRC with an error rate of 3.6% lower than the previous year's results. The proposed approach suggests adding a residual block to mitigate the degradation of performance when adding more hidden layers. In residual blocks, a direct connection, known also as a skip connection, is added. This connection skips some layers and their output is added to the output of those layers.

Classification evaluation metrics

There are different metrics that are used to evaluate the performance of the classification network. Before listing the frequently used metrics, it is necessary to define the following terminologies that are used in the evaluation:

- **True Positive (TP)**: Positive sample classified as positive.
- **False Positive (FP)**: Negative sample classified as positive.
- **False Negative (FN)**: Positive sample classified as negative.
- **True Negative (TN)**: Negative sample classified as negative.

The previous terms are used to define the following performance evaluation metrics:

1. ***Accuracy***: It shows the percentage of the correctly classified samples among all other possibilities:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + PN} \tag{2.2}$$

2. ***Precision***: A representation of the percentage of the true positive samples among all positive predictions:

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

3. ***Recall***: Recall score indicates the percentage of the true positive samples among all actual positive samples. The same score is referred to as *Sensitivity*:

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

4. ***Specificity***: The specificity indicates a percentage of the true negative samples among all actual negative samples:

$$Specificity = \frac{TN}{TN + FP} \tag{2.5}$$

5. ***F1-Score***: The F1-score indicates the harmonic mean of combined precision and recall values. The combined score presents a balance between both quantities and is useful with cases of imbalanced datasets.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.6}$$

2.2 Uncertainty Quantification

Uncertainty quantification is a prominent field that aims to enhance the reliability and robustness of models during deployment. It provides ways to pinpoint sources of uncertainties, and address them through probabilistic or statistical approaches. Generally, two types of uncertainties are often studied: epistemic and aleatoric [30]. The uncertainty of the model prediction owing to a lack of information is represented by epistemic uncertainty.

This type of uncertainty is normally mitigated by exposing models to additional data and domain knowledge. Aleatoric uncertainty, on the other hand, represents the stochasticity and randomness inherent in the data. The uncertainty, in this case, is dependent on the data source and hence is irreducible [31].

2.2.1 Aleatoric Uncertainty

As explained earlier, randomness in data is irreducible, however, different approaches are explored to mitigate the effect of data uncertainty. One of the extensively studied fields in literature is data fusion. As a definition, sensors fusion or information fusion is defined as the process of managing and handling data and information coming from several types of sources in order to improve some specific criteria and data aspects for the decision tasks. The process of fusion consists of combining the outputs of individual sensors, or the outputs of specific algorithms (state vectors and uncertainty matrices) to produce a new combined outcome that is enhanced, extended, enriched, more reliable, more confident, and more robust than those generated by each of the individual sensors, separately. The final goal remains to use redundancies, complementarities, and advantages of a set of data in order to obtain good enough data in order to make the best decision.

Incorporating multiple competitive and complementary sources of data work synergistically to overcome their shortcomings. In general, it is difficult and critical to generate data from a single independent source and use it in a complex application. The reasons are either due to sensor shortage, the nature of the sensed environment, or both of them. Sensors suffer from

several inadequacies and limitations, which can degrade their performances. Some sources of performance degradation are due to drifting errors, where a small offset can lead to a huge error when readings are accumulated over time as in IMU. Additionally, errors can be due to low sensor resolution, surface irregularities, or wheel slipping as in-wheel odometers. Finally, it can be due to uncertainty in readings. Besides the sensors' own imperfections, the sensed environment conditions have an enormous effect on the sensors' output. Sensor noise, for example, disturbs camera images through sunlight intensity and illumination. Similarly, low light at nighttime degrades the output of color cameras as well. Moreover, GPS sensors are affected by an outage in certain areas such as tunnels and forests.

Searching the literature, it has been found that there exist several categorization schemes of sensor fusion. In this section, the most used classes will be listed. The first category is based on the input type that is used in the fusion network, and it includes Data Fusion (Early Fusion), where the fusion takes place at the raw data level. The second category is Feature Fusion (Halfway Fusion), where features are first extracted from each sensor data, and then these features are fused halfway through the network. The last category is Decision fusion (Late Fusion), and in this class, multiple classifiers are used to generate decisions that are then combined to form a final decision. A different categorization was explained by Dasarathy in [32], where he listed five detailed classification classes, including:

1. Data In/Data Out: The input to the fusion network is raw sensor data, and the output is processed (typically enhanced) raw data.

2. Data In/ Feature Out: Raw data is integrated to produce a set of output features.

3. Feature in/ Feature Out: Where the input and output of the fusion network are feature vectors. This class is commonly referred to as feature fusion, symbolic fusion, or information fusion.

4. Feature In/Decision Out: As the name suggests, the input is a feature vector, and the output is a decision. This class is more common in pattern recognition activities where feature vectors are processed to be labeled.

5. Decision in/Decision Out: Where both input and output are decisions and usually referred to as a decision fusion network.

Based on the source of the fused data, sensor fusion can also be categorized as a multimodal fusion [33], where the fused data are obtained from two or more different types of sensors. Fusing LiDAR point cloud and camera images is a good example of this type of fusion, where the two modalities complement the functionality of each other and provide an improved outcome. Another class of this category is multi-temporal fusion, where data are obtained from the same sensor but at different acquisition times. This type of fusion is common in satellite images for monitoring changes on the Earth. The third type of fusion is a multi-focus fusion [34], where images are obtained from different focal lengths. The last class is a multispectral fusion [35], in which images are captured from different wavelength sensors, such as an RGB camera and a thermal camera. This type of fusion is found

in applications similar to pedestrian detection and object recognition. Additionally, based on sensor configuration, sensor fusion can be categorized into a complementary configuration in which two independent sensors are used, and their outputs are combined to complement each other. A perfect example of this type is the fusion of multiple ultrasonic sensors fixed on a robot bumper to expand the area of coverage. Fusion can also be of a competitive configuration (also called redundant configuration), where multiple sensors are used to measure the same property, and the outputs are used for correction purposes as is the case in multiple IMU measurements. The last configuration is the cooperative configuration, where two or more sensors are used to provide an output that cannot be achieved by the individual sensors, such as integrating the outputs of two stereovision cameras in order to get a three-dimensional depth image. Lastly, based on fusion architecture, sensor fusion can be categorized into centralized, decentralized, and hybrid fusion architecture [36]. In a centralized fusion, all data from the different sensors are connected to a central processing unit. After all, data are aligned to a common reference frame, the central unit receives it as one source of information in order to fuse it. In a decentralized fusion, data obtained from sensors are processed locally, then the obtained output is forwarded to a global processing unit for fusion. A hybrid architecture includes sets of data processed locally and forwarded to the global processor, where the remaining data will be processed and fused

Researchers use different combinations of sensors and fuse their reading at different levels in order to compensate for the limitations of the individual sensors. Vision cameras, for example, are essential sensors that generate a

detailed environmental view of the surrounding. They are inexpensive sensors for a given performance (e.g., resolution, accuracy) compared to active ranging sensors, and can provide dense pixel information of the surrounding scene at a relatively low cost. However, normal vision-based systems fail to provide the depth information needed to model the 3D environment. One alternative is to use a stereovision system that consists of multiple cameras with different locations. Nevertheless, these systems are also extremely sensitive to external environmental conditions such as light intensity (low light and direct sunlight) [37] and severe weather situations such as fog [38], snow, and rain. Fusing a vision-based system with a LiDAR, for instance, creates a complementary output that provides depth information while being robust to external weather conditions [39]. The use of infrared and thermal imaging is another active field that researchers often visit for environment perception applications, especially in unfavorable light conditions and night vision. These systems are often used for applications like pedestrian detection and tracking [40],[41],[42] for their ability to detect humans regardless of the light intensity. In the literature, thermal cameras have been fused with either RGB-D [43] or LiDAR sensors [44] to add depth factor and hence improved the system performance; however, this advantage can be dramatically compromised in extreme weather conditions such as high temperatures.

There are several classical algorithms that utilize data fusion for the development of applications that require modeling and propagating data imperfections (inaccuracy, uncertainty). These algorithms apply methods and approaches that are based on the theories of the uncertainties, as illustrated in Figure 2.2 . The following lists a brief compression of these

methods:

- **probabilistic methods:** These methods are based on probability representation for the sensory information. These approaches handle uncertainties in nonlinear systems, however, requires prior knowledge of systems model and data.
- **Statistical Methods:** Utilized to enhance data imputation using a statistical model to model the sensory information. These methods can handle unknown correlations between data, but are limited to linear estimators and have high computation complexity.
- **Knowledge-based Theory Methods:** These set of methods utilize computational intelligence approaches inspired by human intelligence mechanisms. They can handle uncertainty and imprecision in complex nonlinear systems but depends on expert knowledge and extraction of knowledge.
- **Evidence Reasoning Methods:** The models in this category depend on the Dempster combination mechanism. They have the ability to identify conflicting situations, but require high computational cost and require an assumption of the evidence.
- **Interval Analysis theory:** These methods share the operating space in intervals, and are able to handle complex nonlinear systems, but require discretization of the operating space.

, statistical methods, knowledge-based methods (fuzzy logic and possibility), interval analysis methods, and evidential reasoning methods. The variations

of each category are listed in Figure 2.3.

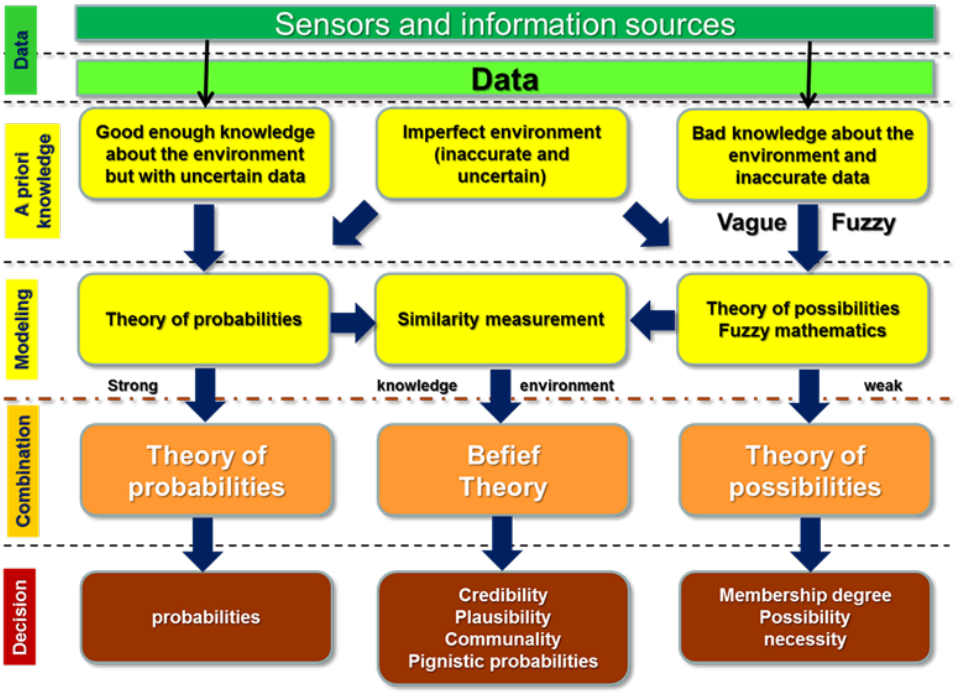


Figure 2.2: Theories of uncertainty for modeling and processing of "imperfect" data

2.2.2 Epistemic Uncertainty

The predicted uncertainty in DL networks is investigated using three primary approaches: the Bayesian approach, the ensemble approach, and the evidential approach. An overview of each approach is highlighted below.

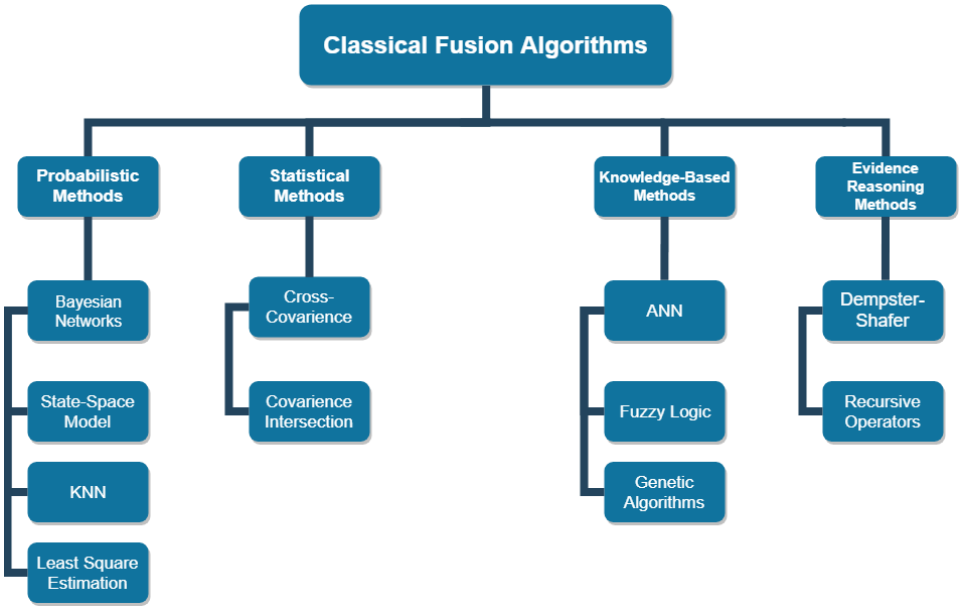


Figure 2.3: Classical approaches for sensor fusion algorithms.

Bayesian Approach

Bayesian Neural Networks (BNN) integrate the principle of Bayesian inference with the standard neural networks. BNN parameters are treated as distributions rather than deterministic point estimates, allowing BNN to represent predictive uncertainty. Let's consider a neural network with parameters ω and training dataset $\mathcal{D}_t = \{(X, Y)\}$. If $p(\omega)$ is the prior distribution assigned to the network weights, then the posterior distribution $p(\omega|x, y)$ is computed by Bayes' theorem as:

$$p(\omega|X, Y) = \frac{p(Y|X, \omega)p(\omega)}{p(Y|X)}. \tag{2.7}$$

At inference, marginalization is applied to predict the label y^* for a test input x^* as:

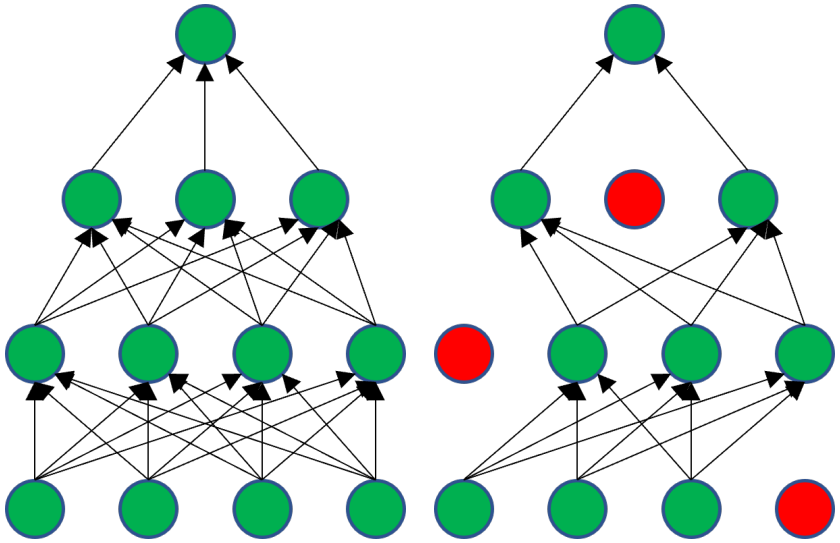
$$p(y^*|x^*, X, Y) = \int p(y^*|x^*, \omega)p(\omega|X, Y)d\omega. \tag{2.8}$$

Calculating the posterior $p(\omega|X, Y)$ requires integrating over all possible values of ω, which makes the process intractable. In most cases, obtaining the posterior is only possible through approximations. The Markov Chain Monte Carlo (MCMC) is one of the popular approximation techniques [45]. Despite its efficacy, MCMC requires drawing multiple samples from the posterior, which makes it computationally costly [46], particularly for deep networks. Recently, Monte Carlo Dropout (MCD) has emerged as a solid solution to the posterior approximation task. Dropout is regularly applied during training to avoid overfitting. The process includes dropping random neurons of the network as illustrated in Figure 2.4. In their paper [7], Gal et al. demonstrated that dropout [47] can be used as a Bayesian approximation; it can be used to predict the network's uncertainty when applied at the inference time.

The simple process to estimate the network's uncertainty is through enabling dropout at inference time, and performing multiple forward passes of the same input. Dropping random neurons during the process allow a variation of the outputs, with a mean and a variance. The mean value represents the prediction, and the variance estimates the uncertainty of predictions. For N multiple passes, the prediction can be represented as the average of all outputs, while the epistemic uncertainty is the variance of the outputs:

$$\tilde{y} = \frac{1}{N}\sum_{i=1}^{N} y^i, \qquad \sigma^2 = Var(y^i) \tag{2.9}$$

Widely adopted in the literature, MCD is applied in a variety of fields to represent the uncertainty estimation of DL models. These applications include classification [48], regression [49], and segmentation [50]. The implementation of MCD is simple and does not require any adjustments to the network's architecture, however, the multiple forward passes at inference introduces an additional computational cost [30]. Moreover, multiple studies indicated that a calibration process is needed to correctly represent the epistemic uncertainty of the network [51, 52].



(a) (b)

Figure 2.4: Structure of a neural network without dropout layer in (a) and with dropout layer in (b).

Deep Ensembles

Deep ensembles include training a number of randomly initialized deep networks on a given dataset and providing a single prediction based on all outputs of the ensemble members. The efficacy of the approach was demonstrated on a wide range of applications, including bioinformatics [53, 54], and image classification [55, 56]. While initially, the purpose of deep ensembles was to enhance prediction accuracy, their ability to effectively estimate predictive uncertainties of deep models has been demonstrated in [6].

Consider a M number of independent networks that map the inputs x to a label y, such that $y = f_{\theta i}(x)$, each network is parameterized by θ parameters, and $i \in 1, ..., M$. The combined prediction of the ensemble can be simply represented as the average predictions of each network at inference:

$$f(x) = \frac{1}{M} \sum_{i=1}^{M} f_i(x) \tag{2.10}$$

Besides averaging, various techniques have emerged to improve the model's performance by better combining the predictions of ensemble members [57]. Weighted average ensembles [58, 59], for example, assign a different weight to each contribution based on the performance of each member on a certain validation subset. Stacking ensembles [60, 61] is another technique in which the contribution of each member is fed to a meta-learner, in order to learn the best combination of these contributions that will lead to improved output. The meta-learner in this case can be any machine learning model such as a simple ANN or a logistic regression model [62]. The weighted average

and stacking ensemble are shown in Figure 2.5 and Figure 2.6 respectively. Two other popular approaches that can be used as ensemble techniques are bagging and boosting. In bagging, which is also known as bootstrap aggregating, subsets of the training dataset are created by combining samples with replacements. each subset is then used as input data to the model. In boosting, on the other hand, the objective is to develop a sequence of models, each of which learns from the mistakes of the one before it. The first model processes a set of samples that are assigned equal weights. Examples that are incorrectly classified are given larger weights than examples that are correctly classified. The weighted samples are then passed to the next model and the procedure is continued until the desired outcomes are obtained.

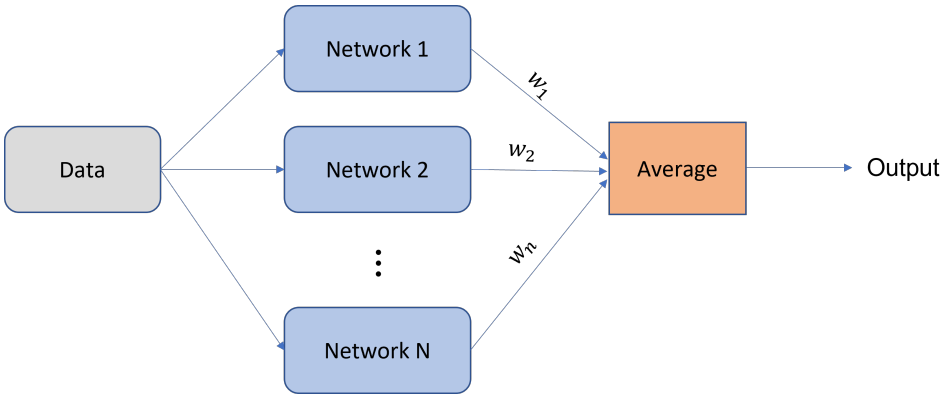


Figure 2.5: Weighted average deep ensemble approach. The output of each ensemble is connected to a learnable weight.

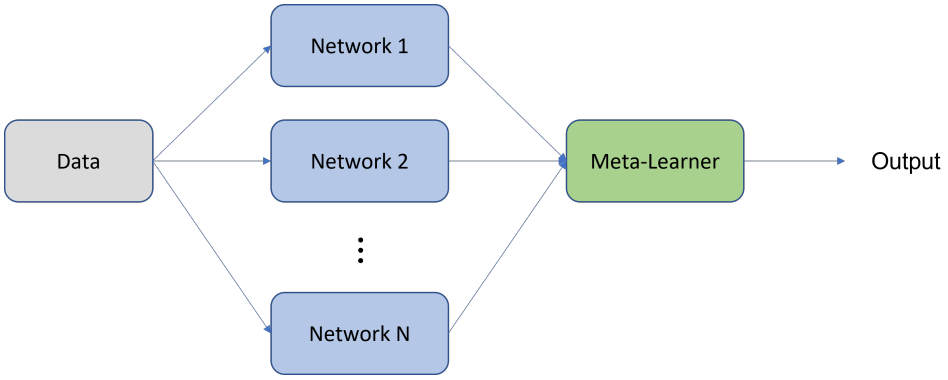


Figure 2.6: Stacking models deep ensemble with a meta-learner approach. The meta-learner can be any machine learning algorithm.

The core concept of ensembles is to combine decisions of different networks and improve prediction accuracy through a decision fusion process. The process is further enhanced when variations are introduced into the behavior of each single ensemble member. This includes different strategies such as random shuffling and random data augmentation of the dataset seen by each model. Moreover, adapting various deep network architectures for each ensemble member is another approach that has been demonstrated to increase the variety of ensemble outputs [63].

Evidential Networks

Evidential networks are non-Bayesian techniques that suggest replacing a neural network's conventional point estimate outputs with a distribution. To achieve that, a higher-order distribution is placed over the likelihood function, while the network estimates the parameters of the distribution. Sampling from the learned distribution yields points that correspond to the

likelihood function and allows for estimating the network's uncertainty.

Evidential networks may be used for classification [8, 64, 65] as well as regression [66]. With different likelihood functions assigned to each task, the goal is to select the higher-order evidential distribution properly. In Bayesian theory, the exact posterior distribution may be obtained if the prior distribution and the likelihood function belong to the same distribution family, i.e. the prior is a conjugate priors [67].

In the classification task, the likelihood function is a categorical distribution with $\mathcal{P}$ probability parameters. Sampling from the categorical distribution yields class labels:

$$y \sim Categorical(\mathcal{P}) \tag{2.11}$$

The conjugate prior of the categorical function is the Dirichlet distribution, where α are the parameters of the distribution, and β is the Beta function:

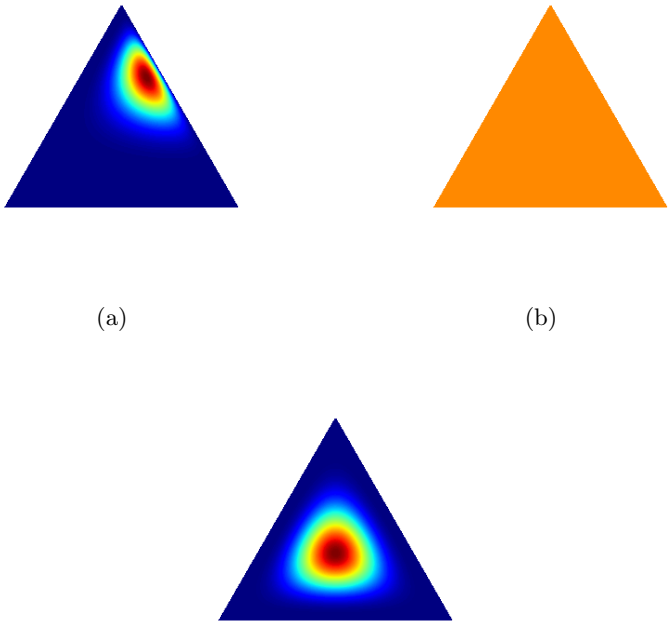
$$Dir(\mathbf{p} \mid \alpha) = \frac{1}{\beta(\alpha)} \prod_{k=1}^{K} p_k^{\alpha_k - 1} \tag{2.12}$$

Different Dirichlet distributions with different α parameters are illustrated in Figure 2.7.

Sampling from the Dirichlet distribution yields the realization of the probabilities $\mathcal{P}$ of the lower-order categorical distribution:

$$\mathcal{P} \sim Dir(\alpha) \tag{2.13}$$

EDL has been used in a variety of applications for estimating predictive uncertainty. In [68], the study proposed a network to predict the uncertainties associated with different attributes in trajectory planning. different from



(a) (b)

(c)

Figure 2.7: Behaviour of a three-class simplex of Dirichlet distribution in different settings. Each case corresponds to a distribution with different parameters: (a) $\alpha = (2, 5, 10)$, represents a confident prediction with low uncertainty (b) $\alpha = (1, 1, 1)$, represents a uniform distribution with total uncertainty, and (c) $\alpha = (5, 5, 5)$ represents a case of high data uncertainty

other implementations that output a single predictive uncertainty value, authors contend that it is crucial to pinpoint the source of uncertainty too. In a different application, EDL was deployed in brain tumor segmentation [69]. The method proposes to perform pixel-level uncertainty estimation for a reliable classification on MRI imaging of the human brain.

In regression, a Gaussian distribution with mean μ and variance σ^2 is used as the likelihood function. Similarly, the target outputs y are obtained by sampling from the likelihood function:

$$y \sim \mathcal{N}(\mu,\, \sigma^2) \tag{2.14}$$

The conjugate prior of the normal distribution is the Normal Inverse Gamma function, parameterized by $(\gamma, \upsilon, \alpha, \beta)$, where $\gamma \in \mathbb{R}$, $\upsilon > 0$, $\alpha > 1$, and $\beta > 0$. the Normal Inverse Gamma function is given by:

$$p(\mu, \sigma^2 \mid \gamma, \upsilon, \alpha, \beta) = \frac{\beta^\alpha \sqrt{\upsilon}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}} \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\Big\{-\frac{2\beta + \upsilon(\gamma-\mu)^2}{2\sigma^2}\Big\} \tag{2.15}$$

Uncertainty estimation has been demonstrated in different regression-related applications. In autonomous navigation, for example, the work in [70] proposed an end-to-end navigation solution based on LiDAR point clouds. The robustness of the system is further enhanced by accounting for the uncertainties of the control action predictions. High levels of uncertainty can be projected by the overall system for predictions linked to OOD scenarios like sensor failures. A fusion network that exploits uncertainty estimation in an object detection scheme is presented in [71]. The network accounts for uncertainties of multiple sensor object detection. As a result of the fusion algorithm and the uncertainty estimation, the system is resistant

to minor sensor displacement, which can affect the accuracy of object detection. The problem is also studied in other fields such as stereo matching. Wang et al. [72] proposed to employ an evidential regression algorithm to the CNN networks. The aim is to detect predictive uncertainty for the resulting disparity map in stereo matching. The effectiveness of the proposed framework was evaluated on cases that include blurred images, and strong light conditions. These situations are very common in self-drive applications and require networks to be aware of cases when predictive uncertainty is high. In [73], evidential regression is deployed to estimate the uncertainty of the human pose estimation from monocular videos. The network takes a 2D pose sequence from an RGB camera and outputs the relative 3D pose coordinates. The task of pose estimation faces multiple challenges such as depth ambiguity, and occlusion. Modeling the depth predictive uncertainty estimation is vital for accurate pose estimation. Other attempts include blood glucose predictions [74], medical imaging [75] and molecular modeling applications [76].

2.3 Out-of-Distribution Detection

Over the last decade, DL models have demonstrated tremendous potential in various fields including computer vision and perception systems. The field relies on the ability of DL models to process high dimensional data and perform predictions and has been shown to outperform human capabilities for certain tasks [77]. Early research in DL-based classification focused on advancing the classifier accuracies, a metric that is often used to benchmark

the model's performance. For real-world implementations, classification accuracy alone does not guarantee the robustness of the system, needed for efficient practical deployment [78]. The challenge is further exacerbated for safety-critical applications such as self-driving vehicles [1, 79] and medical diagnosis systems [80]. In such practical settings, it is hard to ensure that systems are only subjected to "familiar samples". Systems should be able to identify and handle unknown inputs for safe operation.

Consider classification networks, for example. the objective is to determine the best mapping function $f : x \rightarrow y$. This function should maximize the number of correctly assigned labels y to input samples x. Finding such optimal mapping requires adjusting the network's parameters to minimize a loss function ℓ, over all samples of training dataset $\mathcal{D}_{train} = \{x_i, y_i\}$, drawn from a distribution $\mathcal{P}$. The performance of the trained network is evaluated on samples of a testing dataset with independent and identical distribution, such that $\mathcal{D}_{test} \sim \mathcal{P}$. In practical applications, the assumption is too tight and can not be guaranteed for all different samples. A degradation in the model performance is expected for cases of $\mathcal{D}_{inference} \sim \mathcal{P}'$. In machine learning, samples taken from a different distribution than the training distribution are known as OOD samples.

In literature, the term "OOD" is generally used to represent the distribution mismatch between the training and testing data. The mismatch is commonly quantified as either a covariate shift or a semantic shift. well-structured and generalized categories of OOD examples are provided in [81]. An overview of both categories along with their corresponding examples is provided in this section.

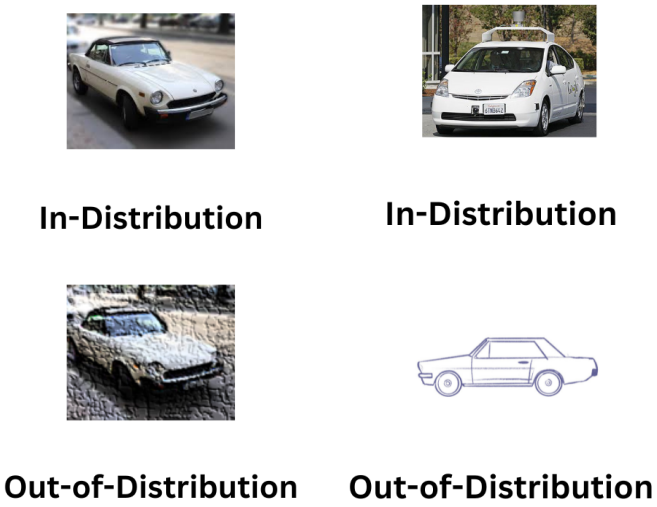
2.3.1 Covariate Shift Out-of-Distribution Data

In covariate shifts, discrepancies in data distributions occur due to a shift in the data domain. The labels, however, remain unchanged. Examples of covariate shifts are adversarial attacks [82], where small perturbations are added to the input data in order to trick the network's decision. The second example of covariate shift is dataset style change [83], where the testing dataset has a different presentation style (i.e. images are represented as sketches or paintings, for example). The third example of covariate shift is when two datasets are captured by different sources. Examples of this category are common in the medical imaging field [84]. In all covariate shift examples, models are expected to generalize and recognize shifted samples in all domains, and consequently categorize them accurately [85]. Illustrations of different examples of covariate shift are shown in 2.8.

Unsupervised domain adaptation (UDA), a category of transfer learning [86], is one of the fields that recognizes the issue of covariate shifts. The main objective of UDA is to create enhanced models that can use data and their labels from a specific domain, and generalize to unlabeled data of other domains. Emerging approaches of UDA are categorized into two main categories. The first category includes domain alignments, where different distance metrics are used to measure and minimize the discrepancy between domains. [87],[88][89],[90]. The second category consists of methods that uses deep neural networks such as adversarial network methods [91], convolutional networks [92], and autoencoders [93].

Domain generalization (DG) is another group of approaches that focus on improving the performance of the networks on OODs with covariate shifts.

In contrast to DA approaches methods in DG assume no access to data in the test domain. Work in DG includes data augmentation methods [94, 95], ensemble approaches [96], and meta-learing approaches [97].



(a) (b)

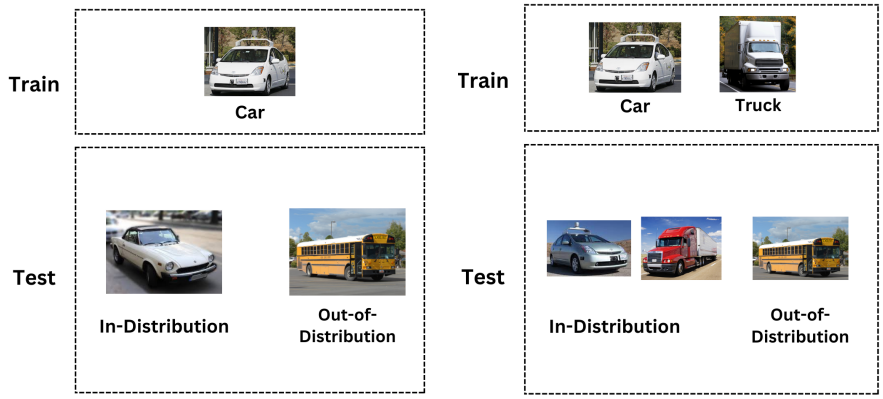
(c)

Figure 2.8: Examples of covariate shift OOD samples due to (a) Adversarial perturbations added to the original image, (b) change in image style, and (c) change of sensor model

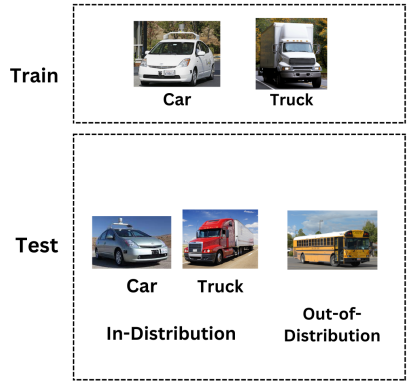
2.3.2 Semantic Shift Out-of-Distribution Data

In semantic shift, OODs refer to data with new labels that have not been part of the training process. In this case, networks are anticipated to reject such OODs without compromising the model's overall accuracy on ID samples. Examples of this type include Open Set Recognition (OSR) [98]. In OSR, the task of the classifier is extended to detect any new unknown sample that was not part of the training data. The classifier is also expected to maintain a reliable classification accuracy on the original ID classes. illustration of the OSR setting is shown in Figure 2.9(a). The second category of semantic shift OODs is known as Novelty Detection (ND) [99]. There are two types of ND: one-class ND and multi-class ND. The classifier in the former is trained on images with a single label, whereas the classifier in the latter is trained on images with multiple labels. For both categories, the classifier should be able to recognize all known images and classify them as ID samples and reject all unknown images as OOD samples. Figures 2.9(b) and 2.9(c) depict both novelty detection categories.

Multiple methodologies are used to detect OOD samples. A baseline of OOD detection was first introduced in [100]. The outputs of SoftMax probabilities were used to detect OODs as they tend to have smaller probabilities compared to correctly classified ID samples. The idea was further extended in [101], where the authors proposed adding a temperature scaling parameter and small perturbations to inputs. The new approach further boosted the distance between SoftMax scores of OOD and ID samples and hence enhanced the detection. Despite the significant performance presented in both studies, both approaches require access to OOD data for training and



(a) (b)



(c)

Figure 2.9: Examples of semantic shift OOD data, including both open set recognition and novelty detection. (a) One-class novelty detection example (b) Multi-class novelty detection example and (c) Open set recognition example.

tuning, a shortcoming that limits the deployment of these methods. In [102], a modified split confidence score along with additional input pre-processing are introduced. The proposed strategies eliminated the need for tuning on OOD datasets. Alternatively, multiple studies [103],[104],[105] proposed using novel loss functions as opposed to the conventional entropy loss for OOD detection.

Chapter 3

Epistemic Uncertainty in Perception: An Evaluation Study

For the past decade, the focus in DL-based object classification research has been to enhance the accuracy of different algorithms. A multitude of works has been presented that propose innovative updates to the classification and training processes [106], often leading to gradual improvements in accuracy. These improvements shape today's powerful DL-based methodologies. Widely accepted in literature, accuracy is a vital metric to evaluate DL models [107] but, does not fully assure the system's robustness [78]. In a more practical setting, an uncertainty estimation in classification output, combined with the prediction probability might represent a more realistic metric. Specifically, in real-world applications, it is hard to ensure that the system is only subjected to inputs from the original dataset [108]. Accurate uncertainty estimation for inputs outside the training domain can help improve robustness. The problem is further exacerbated in safety-critical fields like medical applications and autonomous driving.

In this chapter, the EDL framework is formulated and deployed through an image classification task to quantify predictive uncertainty. During deployment, perception systems are continuously exposed to unrecognized samples. In more challenging cases, these samples share features with the original images seen during training. To cover a full spectrum evaluation of EDL, the selection of relevant OOD datasets is essential. A statistical similarity measure is presented in Section 3.2, and used to evaluate the similarity between the ID classes and the different categories of the proposed OOD samples. Furthermore, a detailed description of the experimental settings and the results of the evaluation are defined and explained in Section 3.3. The presented study evaluates the performance of uncertainty quantification on ID samples, OOD samples, and a combination of both. The objective is to indicate if UQ methods are adequate and efficient in detecting OOD samples that are similar to the ID samples.

3.1 Evidential Deep Learning Framework

In traditional CNN networks, label probabilities are generated by applying a SoftMax activation function to the last layer. The objective is to maximize the likelihood function by minimizing a metric of loss, often a cross-entropy loss, which tends to make the prediction "overconfident" in itself [109]. The output of the SoftMax operation reflects discrete point estimates of class probabilities without proper indication of the uncertainty present in the estimate. EDL on the other hand accounts for the uncertainties in prediction by estimating output distributions instead of rigid labels

[8].

Subjective Logic and Theory of Evidence

The Dempster-Shafer theory of evidence and Subjective Logic (SL) provide a well-defined approach to quantify class uncertainties. The approach suggests that for a K class network, with a belief mass b_k assigned to each class label $k \in K$, an overall uncertainty is u can be expressed, such that:

$$u + \sum_{k=1}^{K} b_k = 1 \tag{3.1}$$

The output of the neural network model (before the conventional SoftMax layer, referred to as logits) is trained to represent the evidence vectors e_k. These evidence vectors represent the amount of support on how confident a sample is, given a particular class. The vectors relate to the belief masses and uncertainties as follows:

$$b_k = \frac{e_k}{S}, \quad u = \frac{K}{S}, \quad \textit{where,} \quad S = \sum_{k=1}^{K} (e_k + 1) \tag{3.2}$$

It is interesting to note that the uncertainty value from Equation 3.2 is inversely proportional to S, which represents the sum of evidence and the number of labels. This notes that a zero-evidence vector would yield a belief mass $b_k = 0$, and the maximum uncertainty $u = 1$. This allows for interesting applications e.g., OOD rejection where an input sample can be rejected as an OOD if the uncertainty estimation u is too high, or above a certain threshold th.

Prior Distribution

The likelihood function for classification models learns over a categorical distribution. EDL places a higher order distribution over the parameters of the categorical distribution (label probabilities) to allow for prediction uncertainty estimation u. To ensure a tractable posterior, it is essential to select a prior conjugate distribution, such as the Dirichlet distribution. Parameterized by $\alpha = [\alpha_1, ..., \alpha_K]$, the Dirichlet distribution's density function is expressed by:

$$D(\mathbf{p} \mid \alpha) = \frac{1}{\beta(\alpha)} \prod_{k=1}^{K} p_k^{\alpha_k - 1} \tag{3.3}$$

Where $\mathbf{p}$ is the probability mass function, $\beta(\alpha)$ is the Beta function, p_k is the probability for label k and α_k can be calculated as:

$$\alpha_k = e_k + 1 \tag{3.4}$$

The probability of each class is equivalent to the expectation of the Dirichlet distribution:

$$\mathbb{E}[D(\alpha)] = \frac{\alpha_k}{S} \tag{3.5}$$

The S here is taken from Equation 3.2 and can be referred to as the strength of the Dirichlet distribution.

Loss Function

The loss function of the training process can be adjusted to learn the evidence vectors that can then be used to calculate the parameters of the

Dirichlet distribution. There are different loss functions that can be implemented. The sum of square error loss function L for a sample i as defined as:

$$\mathcal{L}_i(\theta) = \sum_{j=1}^{K}(y_{ij} - \frac{\alpha_{ij}}{S_i})^2 + \frac{\alpha_{ij}(S_i - \alpha_{ij})}{S_i^2(S_i + 1)} \tag{3.6}$$

A Kullback-Leibler (KL) divergence is used to regularize the prediction of the network. The term measures the similarities of the predicted distribution to a uniform Dirichlet distribution with $\alpha_k = 1 \ \forall \ k \in K$ or zero evidence. The KL term is added to the loss function, and therefore, penalizes samples that deviates from class categories. The KL term is given by:

$$KL[D(p_i \mid \tilde{\alpha}_i) \mid\mid D(p_i \mid 1)] = \\ \log\left(\frac{\Gamma(\sum_{k=1}^{K}\tilde{\alpha}_{ik})}{\Gamma(K)\prod_{k=a}^{K}\Gamma(\tilde{\alpha}_{ik})}\right) + \sum_{k=1}^{K}(\tilde{\alpha}_{ik} - 1)\Big[\psi(\tilde{\alpha}_{ik}) - \psi\Big(\sum_{j=1}^{K}\tilde{\alpha}_{ik}\Big)\Big] \tag{3.7}$$

Where $\tilde{\alpha}_{ik} = y_i + (1 - y_i) \odot \alpha_i$ is the modified Dirichlet parameter. Γ is the Gamma function and ψ is the Digamma function.

3.2 Relevant Out-of-Distribution Samples

The selection of the OOD samples is an essential step for an inclusive evaluation of uncertainty-aware networks. The emphasis is on selecting proper OOD samples to best mimic realistic challenges during inference. In real-life applications, models are frequently exposed to OOD samples that are similar to the ID samples. Hence, it is necessary to explore the performance of the network against samples with semantic similarity. The

representation shift proposed in [110] is adopted to quantify the statistical similarity between any two selected datasets.

In CNNs, distinctive sets of features are learned by the different layers of the network. These features are relevant to the samples of a given dataset leveraged at training. Once the training is completed, the values of the activation feature maps are utilized to form a continuous distribution that represents the latent feature space as learned by the network. Passing a second dataset to the same trained network would result in a different distribution. With these two distributions, it is possible to indicate the level of similarity between any different datasets. The distance between the two distributions can simply represent a similarity measurement R. Ideally, R tends to zero when the network is able to learn similar interpretations of the datasets, i.e. the samples of the two datasets are similar. Figure 3.1 demonstrates the concept through a CNN network that is trained on a set of ID samples. An intermediate layer is selected to extract relevant features, and the activations of that layer are utilized to form a distribution that represents the ID sample distribution, as represented by the latent space of the network. Next, a different dataset is passed through the same network, and the generated distribution from the same layer is compared to the original distribution. The difference between both distributions indicates the amount of shift between the two datasets.

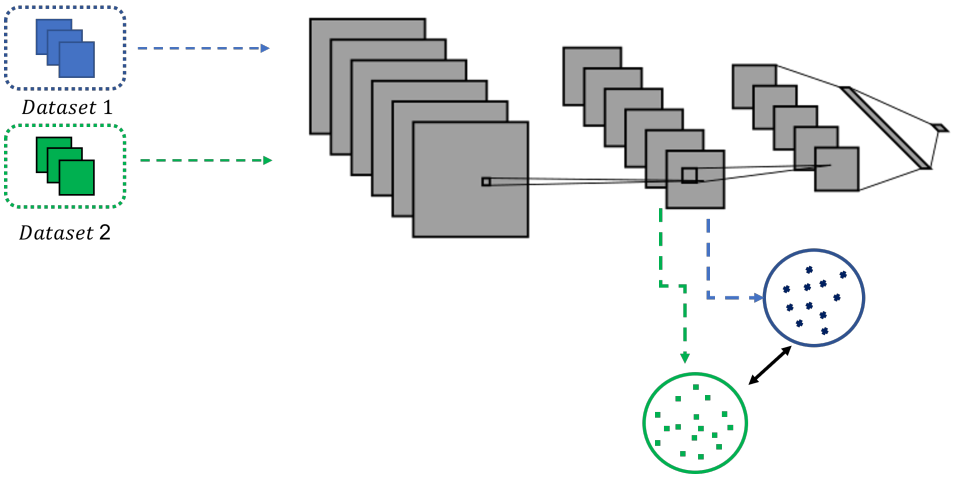


Figure 3.1: The Figure illustrates the process of measuring the distributional shift between two datasets as interpreted by the latent space of a neural network, extracted from one of the intermediate layers.

Consider a CNN model M_θ with $\{L_1, \cdots, L_N\}$, where N is the number of layers of the network, trained on a source dataset $D^S = \{X_1^S, \cdots, X_n^S\}$, where X_i^S is the i^{th} sample of the source dataset. The activations $\phi_n(x) = \{\phi_{1n}(x), \cdots, \phi_{Kn}(x)\}$ are extracted from the n^{th} layer, for all the K filters. The average activation value for each dataset sample for a single filter of hight h and width w is obtained by:

$$c_{lk}(x) = \frac{1}{h \cdot w} \sum_{i,j}^{h,w} \phi_{lk}(x)_{i,j} \tag{3.8}$$

The average activations $c_{lk}(x)$ of all the n samples in the source dataset D^S form a continuous distribution P_{clk}^S over all those samples. Similarly, passing a target dataset through the network would result in different feature realization and hence different activation values. The target activation values

would also form another continuous distribution P_{clk}^{T} of all the m samples of a target dataset $D^T = \{X_1^T, \cdots, X_m^T\}$. The distance between any two distributions can be calculated by the Kullback-Leibler (KL) divergence:

$$\mathcal{D} = KL(P^S \| P^T) = \sum_{x \in \mathcal{X}} P^S(x) \log \left(\frac{P^S(x)}{P^T(x)} \right) \tag{3.9}$$

The similarity metric R between any two datasets, i.e. (D^S, D^T) can be defined based on the average of all the distributions discrepancies $\mathcal{D}$ obtained from all the K filters of a specific layer l, as:

$$R = \frac{1}{K} \sum_{i=1}^{K} \mathcal{D}(P_{clk}^S, P_{clk}^T) \tag{3.10}$$

It is worth mentioning that other distance metrics can be used in substitution of the KL divergence for the calculation of the similarity metric R. These include the Jensen-Shannon divergence, which measures the similarity between two probability distributions and is calculated as the average of the KL divergences between each distribution and the average of the two distributions :

$$D_{JS}(P \| Q) = \frac{1}{2} \left[D_{KL} \left(P \| \frac{P+Q}{2} \right) + D_{KL} \left(Q \| \frac{P+Q}{2} \right) \right] \tag{3.11}$$

and the Wasserstein distance:

$$\mathrm{W}(P, Q) = \inf_{\gamma \in \Pi(P,Q)} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) \, d\gamma(x, y) \tag{3.12}$$

where P and Q are the two probability distributions being compared, c(x,y) is the cost function for transporting a unit of mass from x to y, and γ is a joint probability distribution on X and Y whose marginals are P and Q, respectively.

Figure 3.2 shows samples of the MIO-TCD dataset [111] used to train a CNN classifier. The dataset contains unprocessed raw images collected from street surveillance cameras at different times of the year and reflects multiple objects varying in shape, size, color, type, and orientation to represent a real-world scenario. The samples of the ID dataset include 4 classes of different vehicle types (i.e. Cars, Busses, Trucks, and Pickups). The OOD images, on the other hand, include samples from classes that were not used in training the network. Intuitively, it would be harder to identify OOD samples that are the most similar to ID data. To allow for a systematic understanding of the effect of this similarity, the OOD dataset is categorized into three distinct categories (i.e. pedestrians with low similarity, cyclists with moderate similarity, and trailers with high similarity). The categories are initially grouped based on visual similarity and are further verified via the statistical similarity measurement. Figure 3.3 depicts the three OOD categories used for evaluation.

The statistical similarity between samples of the ID dataset against each category of the OOD datasets is evaluated. Figure 3.4 presents the distributions obtained from the source dataset along with the three categories of the OOD dataset. For visualization purposes, the activations are extracted from filters of different convolution layers in the network. It can be observed that OODs of category 3 (trailers) have distributions closer to the source distributions, while the two other categories (pedestrians and cyclists) result in relatively different distributions. The numerical evaluations of the statistical shift obtained by Equation 3.10 are shown in Table 3.1. From the results, The selection of the categories of the OOD dataset is observed

Figure 3.2: Sample images from the MIO-TCD Dataset. The dataset contains raw and unfiltered images of different vehicle types.

through the average discrepancy metric. It can be concluded that the third category shares more relative features with the ID dataset, compared to both the first and second categories. As stated, the selection of the OOD dataset samples is vital for a comprehensive evaluation of the OOD detection algorithm.

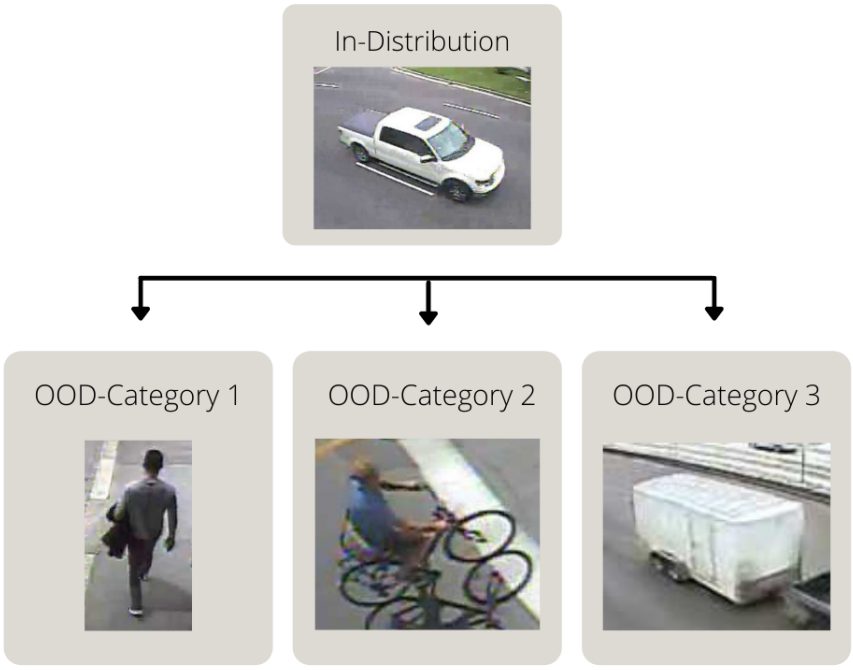


Figure 3.3: Examples of the three categories of OOD samples. Category 1, 2, and 3 refer to classes with low, moderate, and high feature overlap respectively

Table 3.1: *R*-Shift values between the source dataset and the three categories of the OOD datasets evaluated at different layers of the CNN network

Category	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Avg R-shift
Pedestrian	1.5	0.95	0.84	1.15	1.09	1.11
Cyclists	1.03	0.79	0.75	1.05	1.51	1.03
Trailers	0.04	0.05	0.07	0.09	0.10	0.35

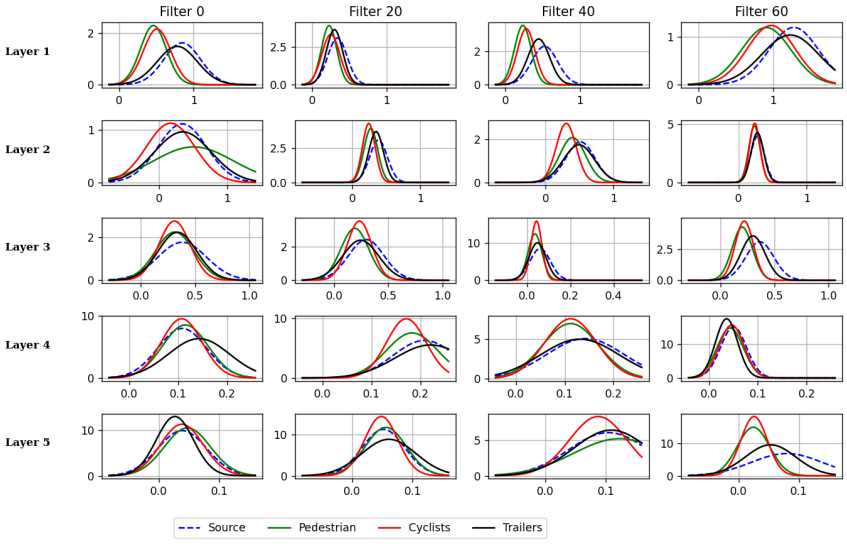


Figure 3.4: Continuous distributions of activation values of source and OOD datasets, obtained from several filters of different layers of the network. The source distribution refers to the ID dataset used to train the network, while the OOD datasets refer to the three categories selected for evaluation

3.3 Threshold-based OOD detection

Identifying OOD samples is an essential capability of uncertainty-aware networks. In Ideal situations, it is desirable that models would fully differentiate between OOD and ID samples. To analyze the same, the behavior of an EDL-based image classifier is evaluated on both ID and OOD samples. The focus of this section is to investigate the effectiveness of applying a threshold on the predictive uncertainty and indicate the shortcomings by studying the classification accuracy, with the presence of both the ID and OOD samples.

For the task of ID classification, The EDL network is trained on the selected classes of the MIO-TCD dataset, illustrated in Table 3.2, using the Stochastic Gradient Decent (SGD) optimizer with a learning rate of 0.006 and a momentum of 0.9 for 30 epochs. During training, the evaluation loss of each epoch is monitored, and the initial learning rate is dropped by a factor of 0.1 when the loss is not reduced for three consecutive epochs. The model achieves a training accuracy of 95.8% and a validation accuracy of 92.4%. The training and validation accuracies are shown in Figure 3.5, while the training and validation losses are shown in Figure 3.6.

Table 3.2: Number of samples for training, validation, and testing splits

Class	Training	Validation	Testing
Truck	9000	1000	1000
Car	9000	1000	1000
Bus	8382	931	1000
Pickup	9000	1000	1000

A confusion matrix is constructed on the testing dataset, which contains 4000 images of ID classes. The inference accuracy is 92.64%. The per-class recall, precision, and misclassification rates are highlighted in Figure 3.7.

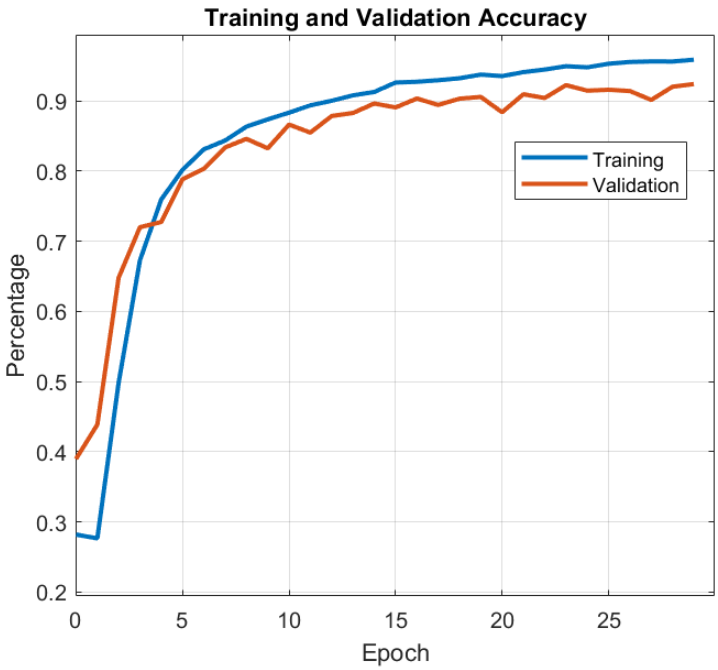


Figure 3.5: Training and validation accuracy

Figure 3.7: Confusion matrix of the classifier with recall, precision, and misclassification rates

Figure 3.6: Training and validation loss

The uncertainty values for any given instance can directly influence the OOD classification. Figure 3.8 shows the outputs of EDL algorithm for different ID and OOD samples. It can be observed that the first ID image is classified as a car, with an uncertainty value that is relatively low compared to three OOD images. Selecting a proper threshold value that would generalize over the full dataset is not trivial. The process gets more challenging as the complexity of the dataset and the number of shared features grow. In practical deployments, samples are classified as OODs when their prediction uncertainties are above a pre-defined threshold. Given the high feature correlation between certain OOD and ID samples, the threshold would also affect the networks capability to properly classify IDs. An ID sample with high uncertainty can thus be classified as an OOD.

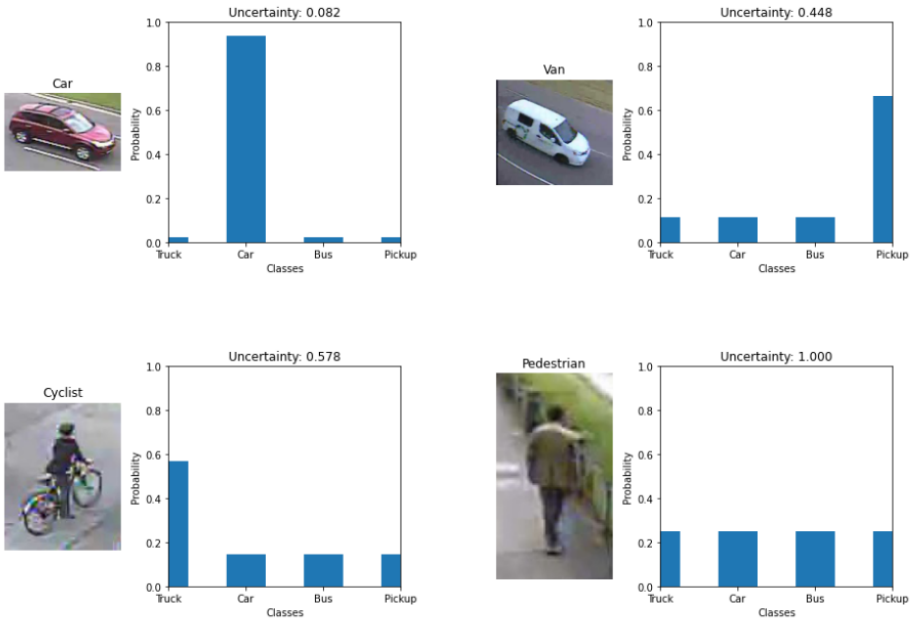


Figure 3.8: Examples of the three categories of OOD samples. The category 1, 2, and 3 refers to classes with low, moderate and high feature overlap respectively

The effect of the choice of uncertainty threshold on ID classification accuracy is illustrated in Figure 3.9. it can be seen that the network achieves the highest accuracy of 92.6%, which corresponds to the original testing accuracy, only when the threshold value is set to 0, indicating that uncertainty does not affect the classification output. As the threshold value increases, more ID samples are rejected, and hence the accuracy of the classifier decreases. The accuracy drops from 92.6% to 42.9% implying that some images of the ID dataset are classified with high uncertainties.

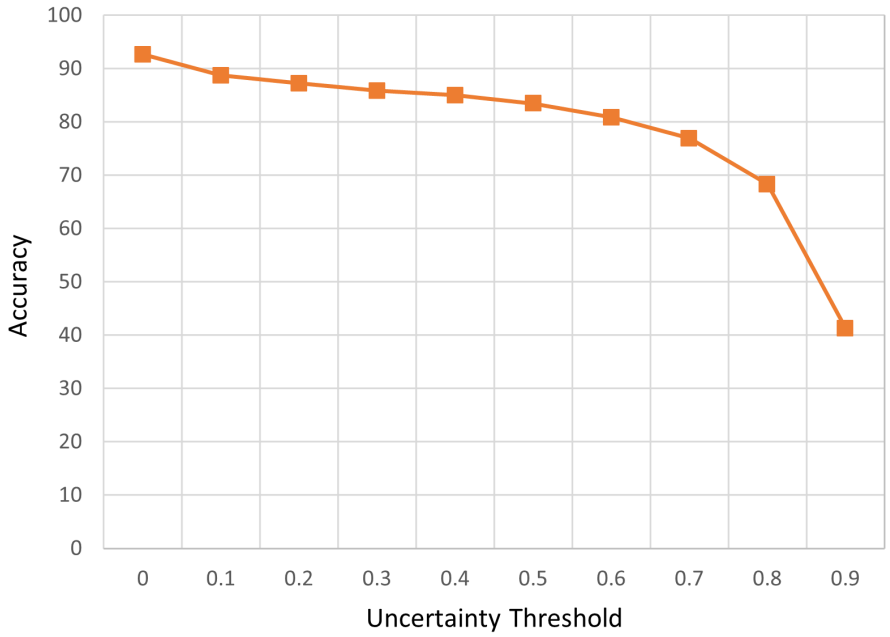


Figure 3.9: Effect of varying the uncertainty threshold On In-Distribution samples of the MIO-TCD Dataset

The analysis is further extended by examining the change in accuracy with OOD samples from the three categories described in Section 3.2. As

hypothesized, Figure 3.10 illustrates the significant variation in accuracy of detecting an OOD sample with different category types e.g., the threshold of 0.7 reports classification accuracies at 87%, 72%, and 55% for Category 1 (low overlap), Category 2 (moderate overlap) and Category 3 (high overlap) respectively.

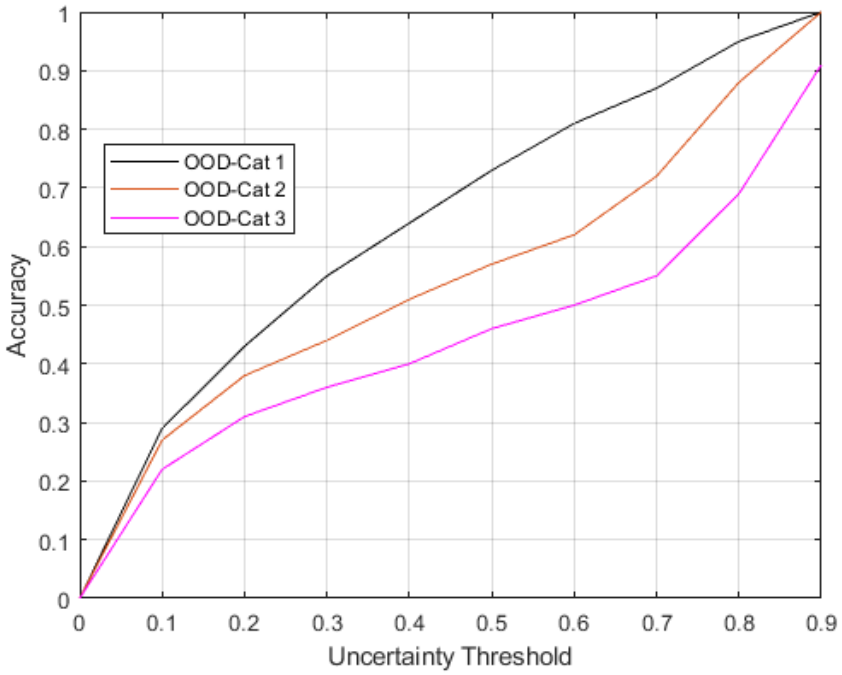


Figure 3.10: Accuracy Vs. uncertainty threshold of samples taken from three different OOD dataset

The threshold selection process needs to reflect the nature of the ID and OOD datasets. An elegant approach to identifying the operating region of the threshold is to visualize the accuracy-threshold relationship. Figure 3.11 shows such a relationship for different OOD categories and ID classification

accuracy. The intersection of the two (at points a_1 and a_2) creates a boundary region over the threshold (points th_1 and th_2) that can be generalized on the entire dataset. Further analyzing the identified operating region for uncertainty threshold bounds, Figure 3.12 illustrates the change in True positive rate (TPR), False Positive Rate (FPR), and ID-OOD combined classification accuracy for varying uncertainty thresholds. The plot highlights the efficacy of the identified region as the TPR and FPR within these regions represent the optimal combination. It is important to note that the combined accuracy reflects the accuracy of the classifier when subjected to data comprising of ID, and all three categories of OOD samples. Figure 3.11 shows that with the increase in threshold, ID accuracy declines while the OOD rejection accuracy improves. This behavior is also seen in Figure 3.12 where the combined accuracy stays consistent with changing threshold, as the increase in OOD accuracy compensates for the decline in ID accuracy.

To summarize, the chapter presents a preliminary study of OOD detection by applying EDL and evaluating its performance on OOD samples. The focus is on using real-world datasets and challenging OOD samples. The choice of datasets poses unexplored challenges on uncertainty-based OOD rejection methods.

Results from the presented experiments indicate that highly overlapped class features impair the separability of ID-OOD samples based on the uncertainty value alone. The overlap causes a loss in the classifier accuracy when a threshold is applied to detect OOD samples. Consequently, it is neither practical nor accurate to operate on a single uncertainty threshold value without compromising on classification accuracy. The findings identify

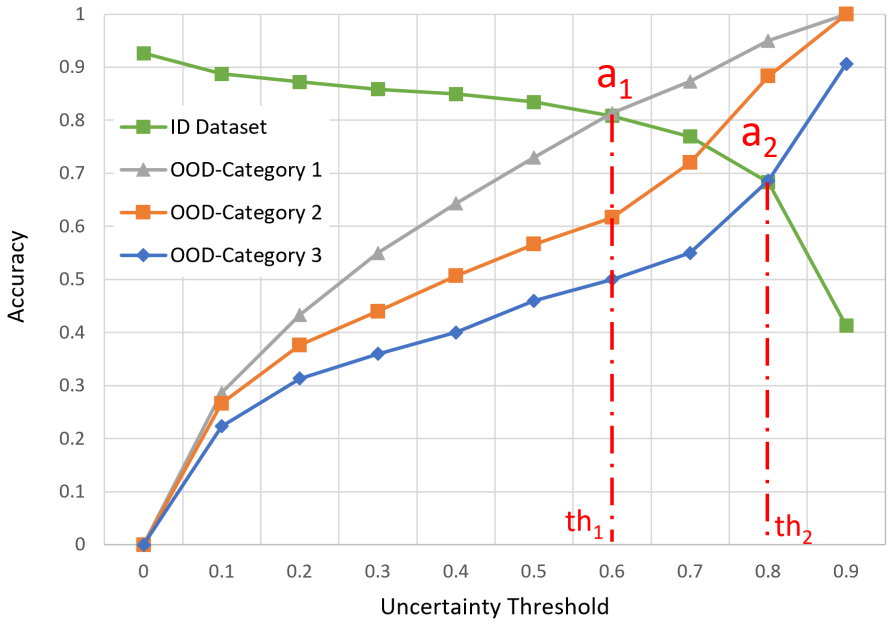


Figure 3.11: Boundary region of the best operating threshold for ID and OOD samples

the need to study further methods that replace the uncertainty threshold, to distinguish ID and OOD samples better while maintaining an adequate overall accuracy of the network.

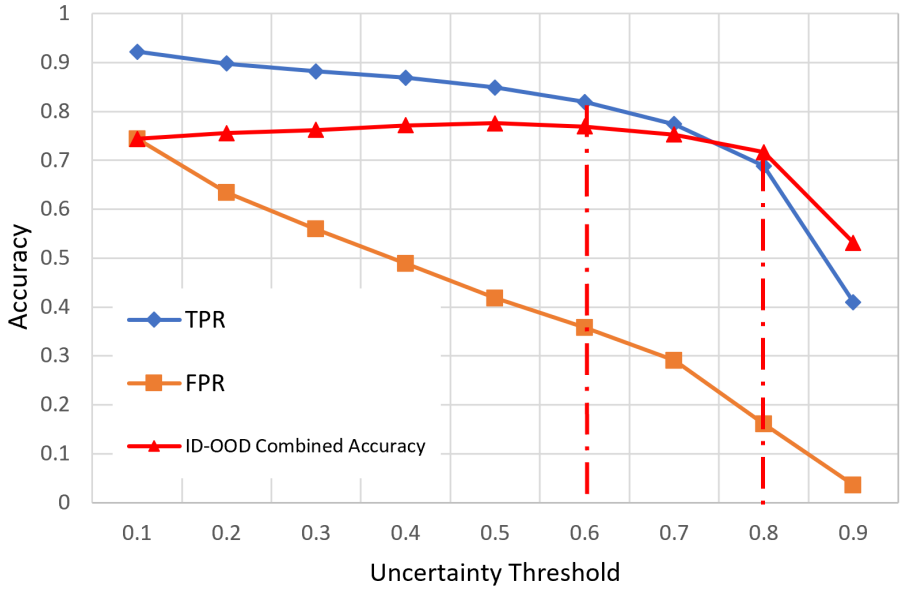


Figure 3.12: True positive rate, False positive rate, and ID-OOD combined accuracy plot

Chapter 4

Dominant Inter-level Feature Exploitation

In machine learning, closed world assumption is heavily relied upon when training models [112] i.e., it is assumed that both the training data and the inference data follow the same distribution. When deployed in a real-world environment, the assumption is too strict and does not allow for efficient transference from training to inference. The discrepancy between the two distributions highlights the need for adaptive models that can robustly tackle OOD data. The shift in the OOD samples can be categorized to be either a covariate or semantic shift when compared to the original training data or ID. Covariate shift refers to OOD data that share similar labels with the ID data. While on the other hand, semantic shift refers to OOD data with completely different labels. In conventional DNNs, the confidence score of the network does not form a reliable measure for OOD detection. The network fails to report lower scores with OOD samples. A more effective approach would involve examining the activation of the network's intermediate layers as they would provide distinct information, which allows identifying dissimilar samples.

This chapter presents an overview of the novel mechanism of DIFE (Dominant Inter-level Feature Exploitation) and further illustrates its application to improve uncertainty-based OOD detection. In section 4.1, the formulation of the framework is detailed. The algorithm of DIFE is applied to a general deep CNN classifier to distinguish between ID and OOD samples, based on their extracted distinctive features. An extensive set of experiments are reported to show the superiority of the proposed framework when compared to other state-of-the-art OOD detection approaches. Section 4.2 highlights the integration of DIFE with an uncertainty-based classification network and explores the possibilities to enhance the detection of OOD samples while maintaining the uncertainty outputs as produced by the original classifier. Experiments illustrate the effectiveness and the ability of the proposed method to distinguish between the ID and OOD samples while significantly outperforming the uncertainty threshold-based OOD detection. In addition, Section 4.3 builds on the concept of DIFE and represents an OOD Detection framework that is based on a distance metric between a distribution of the class-wise ID samples formed by inter-level feature extraction, and another distribution obtained at the inference. Intuitively, an OOD sample at inference will have a larger distance when compared to another ID sample.

4.1 Learning with DIFE

In this section, the novel mechanism of DIFE is presented in detail. The mechanism leverages features extracted from a trained primary net-

work. The mechanism includes sorting and sampling of extracted features to generate a distinct classwise set of features. Moreover, a proposed simple auxiliary network is added to any conventional DL-based classifier. Through the utilization of extracted features, the auxiliary network learns a representation that can effectively distinguish between ID and OOD samples. The implementation of DIFE is realized through the following process:

1. **Primary Network**:

   The primary network refers to any multi-layer deep classifier trained with a conventional supervised learning scheme. The subscript θ is used to denote variables corresponding to the primary network. The primary network model is denoted with M_θ, trained on the dataset D_θ with ground truth labels l_{gt}. The training dataset with K samples for i^{th} class C_i, is denoted as $S_{(1,\ldots,K)}\Big|_{C_i}$. For a test sample $S_t \in D_\theta$,

   $$M_\theta(S_t) := \{\tilde{l}_t\} \tag{4.1}$$

   where, $\tilde{l}_t$ is the predicted label of the sample.

2. **Inter-level Feature Extraction**:

   During classification, each layer of the DL-based classifier performs different levels of feature extractions. Specifically for image-based classifiers, the features extracted by the convolutional layers close to the input are comparable to generic color blobs, while getting increasingly distinct and classwise for the following layers [113]. The concept forms the motivation to distinguish two different images (e.g., an ID and OOD sample) by examining their mid-layer or Inter-level feature

maps, even if the two samples are classified into the same class. The activation map is denoted by $\mathcal{A}(L_m)$, for the m_{th} layer L_m of the primary network with J layers, $m \in [1, \ldots, J]$. For a chosen inter-level layer, the corresponding activation is flattened and stored $\forall S \in D_\theta$. Figure 4.1 represents the primary network and the process of extracting classwise features from an intermediate layer of the network.

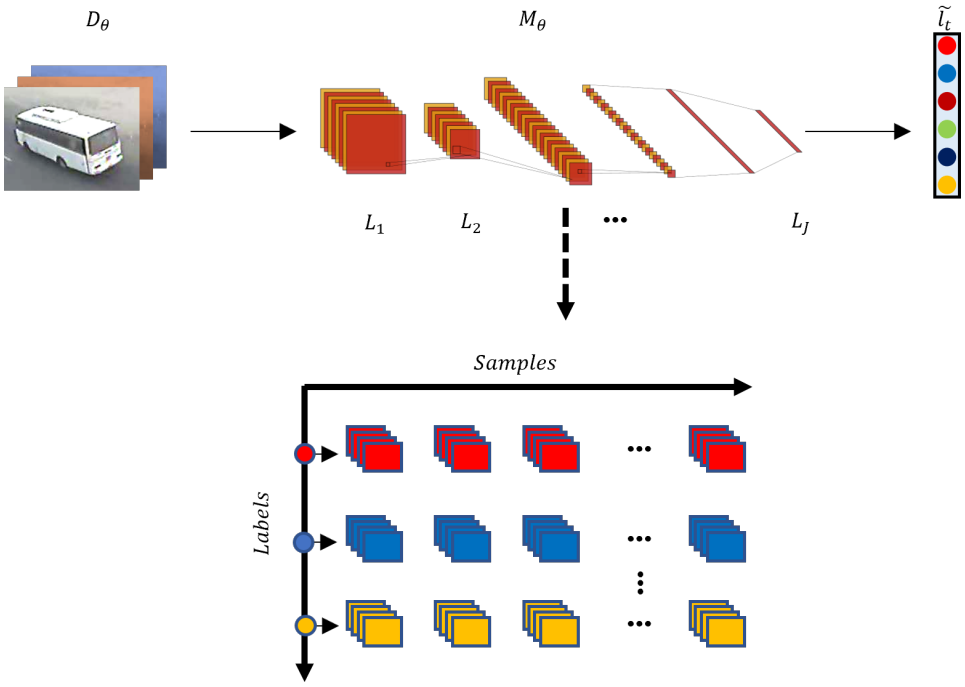


Figure 4.1: Pre-trained primary network classifier with labels (Top). Extracted features from one of the intermediate layers of the primary network (Bottom).

3. **Dominant Feature Selection and Indicative Vector**:

   Classwise samples share similar features, hence their activation maps

are similar. To efficiently differentiate samples, the dominant features in the activation maps are determined. This is achieved by creating class-specific subgroups where each sample in the subgroup is classified into the same class by the primary network. Then, the activation values for the classwise subgroup are averaged. The classwise average activation values are obtained by:

$$A_G\Big|_{C_i} = \frac{1}{|G|}\sum_{g=1}^{|G|}\mathcal{A}(L_m)_g \tag{4.2}$$

Where G is the number of samples of a specific class and $\mathcal{A}(L_m)_g$ are the corresponding activation maps of the class samples.

The average activations A_G are used to determine the classwise indicative vector $V\Big|_{C_i}$, which represents the features that are mostly activated for all samples.

4. **Sampling for Auxiliary Network Dataset**:

   The subscript ϕ to used to denote variables corresponding to the auxiliary network. To create the training dataset D_ϕ for auxiliary network model M_ϕ, values and indices of the classwise indicative vector $V\Big|_{C_i}$ were sorted by importance. To extract the most dominant features from the sorted vector, the top $\zeta\%$ indices were sampled to generate classwise indices $V_{Idx}\Big|_{C_i}$. The classwise indices are then applied to sample dominant features for each sample in D_θ, resulting in a new set of samples $S_\phi = V_{Idx}\dot{\sim}\mathcal{A}(L_m)$. The samples are then paired with their corresponding ground truth labels $\{S_\phi, l_{gt}\}$ to form the auxiliary network dataset D_ϕ. The process of selecting the dominant features

and the corresponding classwise indicative vector is shown in Figure 4.2

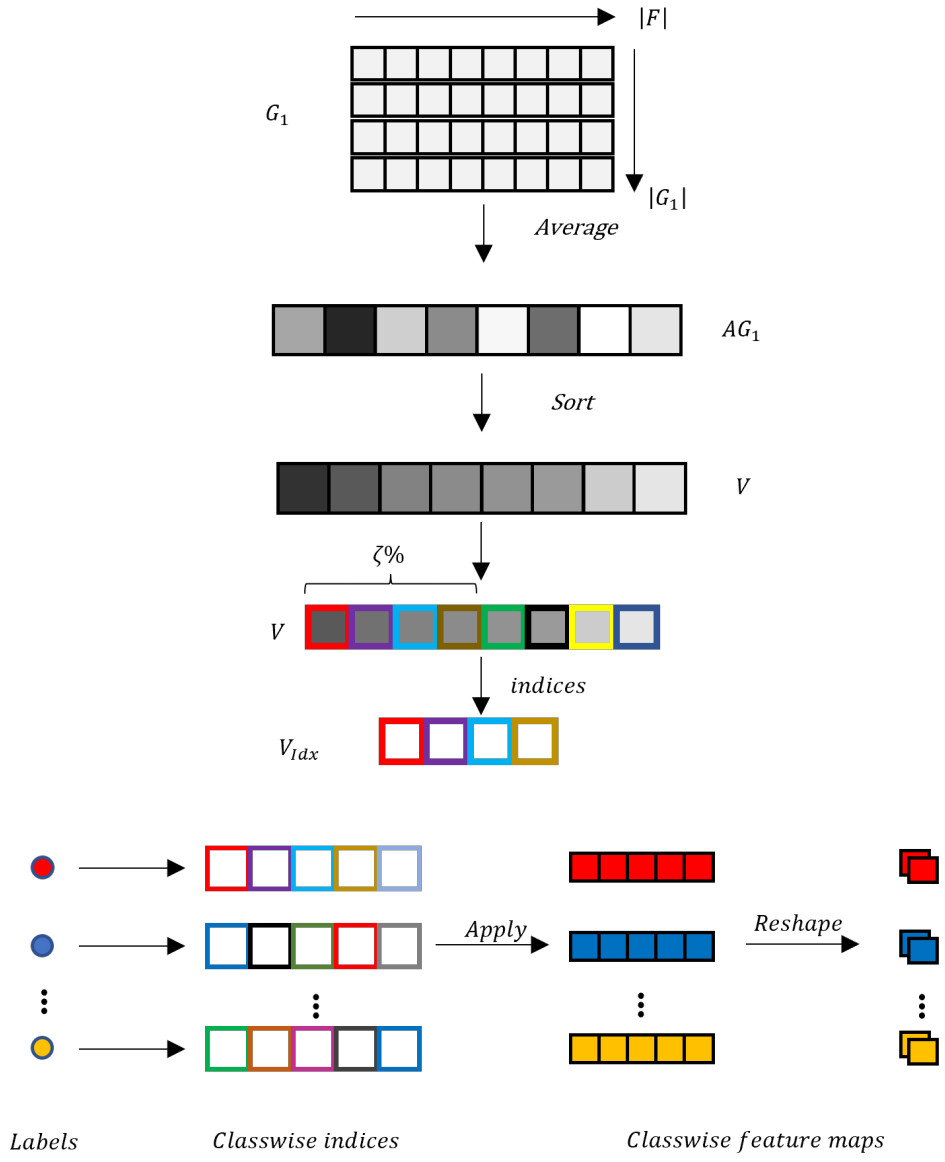


Figure 4.2: Dominant feature selection and the resulting class-wise indicative vector.

5. **Auxiliary Network Training and Inference**:

   M_ϕ is trained on D_ϕ to learn representations that can distinguish ID and OOD samples. At inference, the class-wise $V_{Idx}\Big|_{C_i}$ index vector is applied to the corresponding sample, classified by M_θ, to extract its dominant features. It is then passed to the auxiliary network for classification as S_ϕ. The sample is categorized to be an ID if the classification labels of both networks match and OOD otherwise. The training and inference of the auxiliary network are shown in Figure 4.3

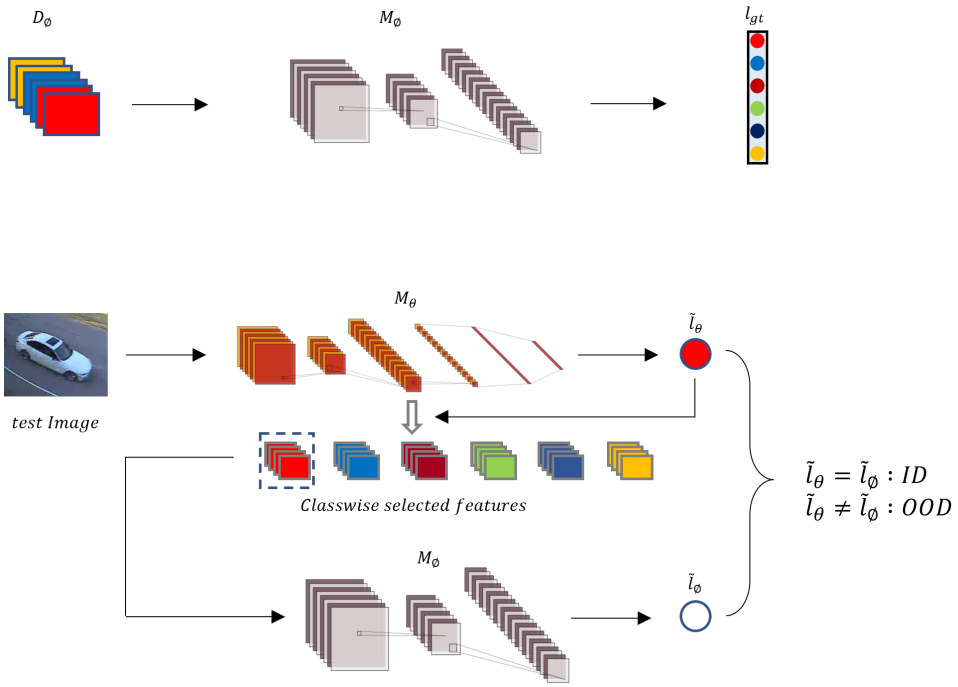


Figure 4.3: Top: Dataset of class-wise dominant features used to train the proposed auxiliary network. Bottom: Auxiliary network during inference.

Performance Evaluation

The performance of DIFE is evaluated on a range of experiments including different network architectures and different evaluation datasets. The implementation details are listed below.

Datasets

The proposed method is demonstrated on three different benchmarking datasets including the MNIST [114], CIFAR-10, and CIFAR-100 [115]. In each case, the OOD detection was evaluated on near-OOD and far-OOD datasets. Near-OOD refers to datasets that share similar semantic features with the original training dataset, whereas far-OOD refers to datasets that are significantly different from the data used at training. For the MNIST dataset, 10 classes representing numerical digits (zero to nine) were selected to train the primary network as ID classes. The near-OOD dataset is the Fashion-MNIST [116] and the NotMNIST dataset, while the far-OOD dataset is the CIFAR-10 and the Tiny ImageNet [117]. In the second case, the CIFAR-10 was selected as the ID dataset and evaluated on both CIFAR-100, Tiny ImageNet as the near-OOD dataset and the MNIST, SVHN [118] as the far-OOD dataset. Finally, the CIFAR-100 was selected as the ID dataset and evaluated on CIFAR-10, Tiny ImageNet as near-OOD, and MNIST, SVHN as far-OOD.

In addition to OOD detection, the framework was also evaluated on the Open-Set Recognition (OSR) task. The MIO-TCD [111] dataset is a large-scale real-world dataset that includes more than 600,000 unprocessed

raw images collected from street surveillance cameras at different times of the year and reflects multiple objects varying in shape, size, color, type, and orientation. Four classes from the dataset were selected as ID classes (labels: Bus, Car, Truck, Pickup) and three from the remaining classes (labels: Pedestrian, Cyclist, Trailer) were used as OOD data. The classes for ID and OOD data were selected to ensure that visually similar and related features appear in both categories to present a more realistic approach to OOD detection.

Networks

Primary Network:

The primary network was trained on three standard CNN architectures including the LeNet, ResNet-18, and VGG16. For a fair comparison with other SOTA approaches, the LeNet network was applied whenever the MNIST was chosen as the ID dataset, whereas the Res-Net-18 was used for both CIFAR-10 and CIFAR-100 being the ID datasets. Moreover, the VGG-16 architecture was applied to evaluate the performance of the framework with the MIO-TCD dataset on the OSR task and to study the sensitivity of varying the hyperparameters of the framework.

Auxiliary Network:

The auxiliary network is trained as a standard CNN model with two conv-relu blocks followed by two fully connected layers and a softmax layer. A cross-entropy loss with 50 epochs and a fixed learning rate of 0.01 is used to train the auxiliary network. The choice of architecture for the auxiliary network is open and can be scaled to cater to the extracted feature dimen-

sions. The results from the proposed mechanism are highlighted in the next section.

Performance of the Auxiliary Network

The features used for training the auxiliary network were extracted from the intermediate Conv layers L_{ext} of the primary network (2^{nd} Conv layer for LeNet and 3^{rd} Conv block of Res-Net). intermediate layers were chosen under the intuitive assumption that the sample loses features as it progresses through the network, while eventually getting matched to a single label. Instinctively, distinctive features that separate an ID and OOD sample would only be captured in these earlier layers. For the training of both the auxiliary networks, the sampling ratio, α was set to 0.5. The effect of varying these hyperparameters is reported in the next section. Table 4.1 compares the performance of the proposed mechanism of DIFE with three SOTA OOD detection approaches including MSP [100], ODIN [119], and KLM [120]. The results report two common OOD detection metrics, the first is the Area Under the Receiver Operating Characteristic curve (AUROC), which illustrates the ratio between the TPR and the FPR. The second metric is the Area Under the Precision-Recall curve (AUPR), which plots the precision and recall curves at different thresholds.

Open-Set Recognition

The performance of DIFE was additionally evaluated on the OSR task, where the OOD samples are a separate subset taken from the same training dataset. Four classes of the MIO-TCD large-scale dataset were used for

Table 4.1: Performance Evaluation of DIFE in comparison with MSP, ODIN, and KLM OOD detection approach. **Bold** represents best scores.

ID	OOD	AUROC	AUPR
		MSP/ODIN/KLM/DIFE (Ours)	
MNIST	Fashion-MNIST	93.7/**94.9**/82.6/93.5	93.6/**94.7**/72.1/92.8
	NotMNIST	89.2/89.9/78.0/**92.2**	77.8/80.8/54.9/**91.7**
	Near-OOD (mean)	91.5/92.4/80.3/**92.9**	85.7/87.8/63.5/**92.3**
	CIFAR-10	98.6/**99.1**/96.5/98.8	98.6/**99.1**/95.6/97.2
	Tiny	98.8/**99.2**/96.9/98.7	98.8/**99.2**/96.0/96.9
	Far-OOD (mean)	98.7/**99.2**/96.7/98.8	98.7/**99.2**/95.8/97.1
CIFAR-10	CIFAR-100	87.1/77.6/78.7/**89.2**	**85.9**/73.2/72.8/84.2
	Tiny	86.6/77.3/79.1/**88.4**	83.1/70.1/70.7/**87.8**
	Near-OOD (mean)	86.9/77.5/78.9/**88.8**	84.5/71.7/71.8/**86.0**
	MNIST	89.9/90.9/82.4/**94.6**	66.9/64.7/40.7/**80.7**
	SVHN	90.9/73.3/86.0/**92.4**	78.2/42.1/70.0/**89.2**
	Far-OOD (mean)	90.4/82.1/84.2/**93.5**	72.5/53.4/55.4/**85.0**
CIFAR-100	CIFAR-10	78.3/78.2/72.5/**80.1**	79.5/79.1/67.2/**81.9**
	Tiny	81.8/81.4/78.4/**85.3**	86.3/85.3/80.5/**87.4**
	Near-OOD (mean)	80.1/79.8/75.5/**82.7**	82.9/82.2/73.9/**84.7**
	MNIST	77.8/83.7/77.6/**90.2**	54.2/62.0/32.9/**86.9**
	SVHN	76.0/71.1/71.3/**84.5**	60.8/52.4/40.3/**77.4**
	Far-OOD (mean)	76.9/77.4/74.5/**87.4**	57.5/57.2/36.6/**82.2**

training and three other classes were utilized as OODs. The OODs are also categorized into near-OODs (Trailer), and far-OODs (pedestrians and cyclists). Results are reported in Table 4.2

Table 4.2: Evaluating the performance of DIFE on the Open-Set Recognition task with MIO-TCD dataset.

ID	OOD	AUROC	AUPR
MIO-TCD	Near-OOD	83.9	81.1
	Far-OOD	92.7	90.4

Sensitivity to parameter change

In this section, we study and report our observations when changing the parameters that define the components of the proposed mechanism and corresponding auxiliary network. These include the effect of 1) extraction layer selection, and 2) sampling ratio. The VGG-16 architecture is used as the primary backbone network and near-OOD datasets are used for the evaluation.

1. **Extraction Layer Selection:** Consider two dissimilar samples that are predicted to be the same label by the primary network. At the output layer, both samples, irrespective of their dissimilarity, are matched to the same label. Features extracted from earlier intermediate layers, on the other hand, exhibit the differences in the sample. This forms the motivation for layer selection in the proposed mechanism. To visualize the effect of layer selection, features are extracted from both the first and last convolutional layers of VGG-16 and used to train the auxiliary network. Results in Table 4.3 suggest that the auxiliary network performs best with features extracted from mid-layers, as the first layers capture low-level features, and the last layers do not contain

distinctive information for different samples.

Table 4.3: Effect of changing the selected extraction layer on the performance of DIFE on OOD detection.

Layer	AUROC / AUPR		
	MNIST	CIFAR-10	CIFAR-100
L_1	75.6 / 74.1	62.3 / 54.7	55.6 / 55.4
L_{13}	68.3 / 66.8	55.4 / 52.8	51.1 / 49.7

2. **Sampling Ratio:** Table 4.4 studies the effect of varying the sampling ratio, thereby varying the size of the feature vector used to train the auxiliary network. Two experiments are reported with extreme α values. With $\alpha = 1$ i.e., no resizing, the results show no improvement in the auxiliary network performance, as expected. Choosing a small sampling ratio of $\alpha = 0.25$ on the other hand is expected to cause a significant loss of information including necessary feature information. The same is evident in the reported results.

Table 4.4: Effect of varying the sampling ratio α on the performance of DIFE for OOD detection.

α	AUROC / AUPR		
	MNIST	CIFAR-10	CIFAR-100
$\alpha = 1.0$	76.4 / 73.6	65.2 / 66.4	52.3 / 51.7
$\alpha = 0.25$	74.1 / 72.3	58.7 / 55.6	52.2 / 49.6

4.2 Efficient Uncertainty-Based OOD detection with DIFE

Uncertainty quantification approaches are promising methods to measure predictive uncertainty estimates of DL models. The detection of OODs based on uncertainty threshold, however, is not reliable. In real-world datasets, highly overlapped class features pose challenges that impair the separability of ID-OOD samples based on the uncertainty value alone. In this section, the concept of DIFE is extended and applied to efficiently recognize OOD samples. A simple auxiliary network utilizes features extracted from the primary network and learns unique representations to distinguish between the ID and OOD data while preserving the assigned uncertainty estimation for each output. In this particular application, the activations of the ID samples are categorized into groups G based on uncertainties for correctly classified samples. It was noted that including samples of both low and high uncertainty values $\{U_{Lth}, U_{Hth}\}$ improves the classification accuracy on the ID dataset. The two groups were generated following the equation below:

$$G = \begin{cases} 1, \text{if } \tilde{U}_t > U_{Hth} \\ 2, \text{if } \tilde{U}_t < U_{Lth} \end{cases} \tag{4.3}$$

The dominant features for groups G_1 and G_2 are determined by averaging the activation values for the samples of each group. The average activations A_{G_1} and A_{G_2}, for G_1 and G_2 respectively, are used to determine the classwise indicative vector $V\Big|_{C_i}$, which represents the features that are exclusively

activated in a certain group and not the other $V\Big|_{C_i}$ is calculated by:

$$V\Big|_{C_i} = A_{G_1}\Big|_{C_i} - A_{G_2}\Big|_{C_i} \tag{4.4}$$

Finally, the top $\zeta\%$ and the bottom $\delta\%$ indices were sampled to generate classwise indices $V_{Idx}\Big|_{C_i}$, which is later used to define the auxiliary network dataset. The modified process is shown in Figure 4.4

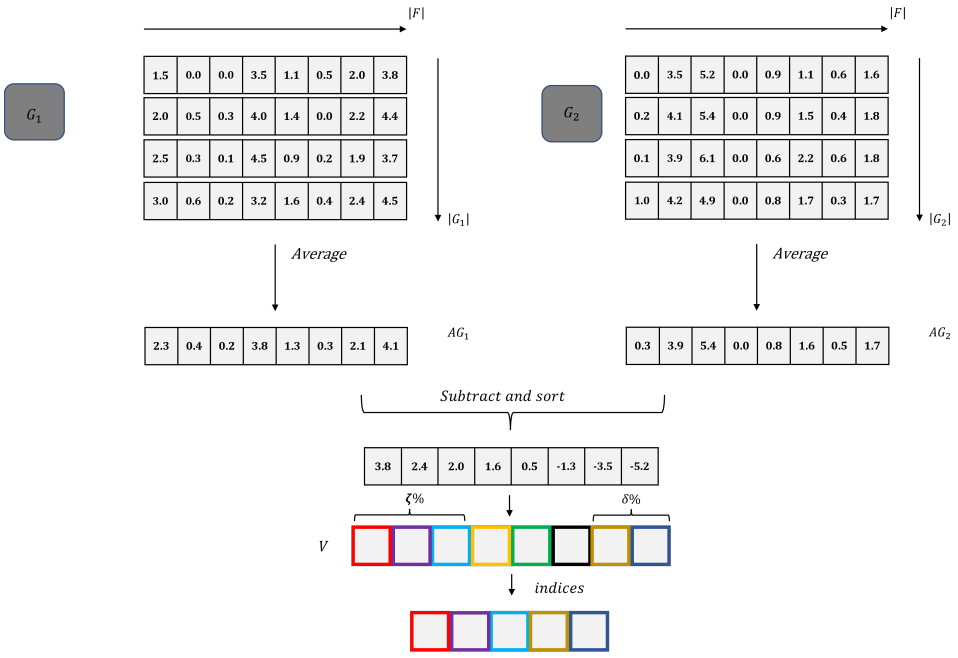


Figure 4.4: Dominant feature extraction for samples of G_1 and G_2 and the corresponding indicative vector

To summarize, the main contributions of the work presented in this section are listed below:

1. A novel mechanism of DIFE is proposed. This mechanism leverages features extracted from a trained primary network. The mechanism

includes grouping, sorting, and sampling of extracted features to generate a distinct classwise set of features.

2. An auxiliary network is developed. This network can be added to any conventional DL-based classifier with uncertainty estimation. Through the utilization of extracted features, the auxiliary network learns a representation that can effectively distinguish between ID and OOD samples.

3. A range of experiments are evaluated on different datasets in support of the proposed method with the aim of extending off-the-shelf networks to real-world and complex datasets for OOD detection.

Dataset

The proposed method is demonstrated on the MNIST [114], CIFAR-10 [115], and MIO-TCD dataset [111]. The MNIST and CIFAR-10 are commonly used datasets for OOD detection benchmarking and MIO-TCD was chosen for its real-world characteristics and noisy images. For the MNIST dataset, 10 classes representing numerical digits (zero to nine) were selected to train the primary network, and the NotMNIST dataset (labels: alphabets A to J) were used as OOD data. For CIFAR-10, the primary network trained on five classes (labels: Airplane, Automobile, Bird, Cat, Deer) and the remaining five (labels: Dog, Frog, Horse, Ship, Truck) were used as OOD data. MIO-TCD, on the other hand, includes more than 600,000 unprocessed raw images collected from street surveillance cameras at different time of the year and reflect multiple objects varying in shape, size, color,

type, and orientation. The conditions imposed represents a real-world scenario, with factors that are often missing from other studied datasets. Four classes from the dataset were selected (labels: Bus, Car, Truck, Pickup) to train the network and the three from the remaining (labels: Pedestrian, Cyclist, Trailer) were used as out-of-distribution data. The classes for ID and OOD data were selected to ensure that similar and related features appear in both categories to present a more realistic approach to OOD detection.

Network Architecture

Primary Network:

The primary network was trained as standard CNNs with a series of 5 conv-relu-maxpool-batchnorm blocks followed by 2 fully connected layers and an output layer. EDL was selected for uncertainty estimation as it has been shown to demonstrate efficient uncertainty estimation [121]. The networks were trained using the sum of squares loss introduced in [8]. EDL is trained and evaluated on each of the three datasets. Previous evaluations indicate high sensitivity to uncertainty threshold, and as a result, the primary network fails to separate the ID and OOD samples effectively leading to a significant drop in overall classifier accuracy.

Auxiliary Network:

The auxiliary network is trained as a standard CNN model with two conv-relu blocks followed by two fully connected layers and a softmax layer. The choice of the architecture for the auxiliary network is open and can be scaled to cater to the extracted feature dimensions.

Performance of the Auxiliary Network

Results from the proposed mechanism are shown in Table 4.5. The proposed mechanism significantly outperforms uncertainty threshold-based OOD detection. The features used for training the auxiliary network were extracted from the intermediate conv layers L_{ext} of the primary network (2^{nd} conv layer for MIO-TCD and 3^{rd} conv layer for MNIST and CIFAR-10). Layers closer to the input layer were chosen under the intuitive assumption that the sample loses features as it progresses through the network, while eventually getting matched to a single label. Instinctively, distinctive features that separate an ID and OOD sample would only be captured in these earlier layers. For the training of both the auxiliary networks, the sampling ratio, α was set to 0.5, with the grouping criteria U_{Hth} and U_{Lth} set as 0.8 and 0.2, respectively.

Sensitivity to parameter change

This section studies and reports observations when changing the parameters that define the components of the proposed mechanism and corresponding auxiliary network. These include the effect of

1. Extraction layer selection.
2. Grouping parameters
3. Sampling ratio.

Extraction Layer Selection:

Consider two dissimilar samples that are predicted to be the same label by

Table 4.5: Performance of classifier on ID (Remaining) detection and OOD detection (Rejected) with both uncertainty threshold and proposed DIFE on MNIST, CIFAR-10, and MIO-TCD datasets.

Dataset	Method	ID	OOD	Average
MNIST	Uncertainty Threshold : 0.1	83.9%	89.45%	86.7%
	Uncertainty Threshold : 0.5	91.9%	71.7%	81.7%
	Uncertainty Threshold : 0.9	93.9%	57.6%	75.75%
	Auxiliary Network (Proposed)	98.2%	92.7%	**95.5%**
CIFAR-10	Uncertainty Threshold : 0.1	66.3%	55.84%	61.1%
	Uncertainty Threshold : 0.5	89.12%	27.8%	58.5%
	Uncertainty Threshold : 0.9	96.12%	10.1%	53.1%
	Auxiliary Network (Proposed)	97.8%	84.5%	**91.1%**
MIO-TCD	Uncertainty Threshold : 0.1	73.7%	70.9%	72.3%
	Uncertainty Threshold : 0.5	89.5%	54.7%	72.1%
	Uncertainty Threshold : 0.9	95.0%	46.1%	70.55%
	Auxiliary Network (Proposed)	99.4%	92.2%	**95.8%**

the primary network. At the output layer, both samples, irrespective of their dissimilarity, are matched to the same label. Features extracted from earlier layers e.g., close to the input layer, on the other hand, exhibits the differences in the sample. This forms the motivation for layer selection in the proposed mechanism where layers closer to the input layer are selected. To visualize the effect of layer selection, experiments on two layers (L_{+1} and L_{+2}) following L_{ext} are reported. Specifically, two additional feature extraction layers were evaluated. Results in Table 4.6 suggest that the auxiliary

network perform best with features extracted from earlier layers.

Table 4.6: Effect of layer selection on classification accuracies of ID and OOD Data.

Layer	MNIST		CIFAR-10		MIO-TCD	
	ID	OOD	ID	OOD	ID	OOD
L_{+1}	92.1%	86.2%	87.8%	74.5%	98.1%	26.7%
L_{+2}	90.3%	83.8%	94.4%	78.3%	99.2%	30.1%

Grouping Parameters:

Table 4.7 highlights the effect of changing the grouping parameters (U_{Hth} and U_{Lth}). The experiments demonstrate that a more selective grouping criteria leads to a stable detection accuracy. Relaxing the thresholds on the other hand, results in a significant drop in OOD detection. We posit that a more selective criteria samples distinct features from the two groups leading to effective classification with the auxiliary network.

Table 4.7: Effect of different grouping parameters on classification accuracies of ID and OOD Data.

Threshold		MNIST		CIFAR-10		MIO-TCD	
U_{Hth}	U_{Lth}	ID	OOD	ID	OOD	ID	OOD
0.9	0.2	97.1%	85.3%	96.2%	82.5%	99.3%	84.3%
0.6	0.4	96.5%	77.3%	94.5%	75.2%	94.6%	44.9%

Sampling Ratio:

Table 4.8 studies the effect of varying the sampling ratio, thereby varying

the size of the feature vector used to train the auxiliary network. Two experiments are reported with extreme α values. With $\alpha = 1$ i.e., no resizing, the results show no improvement in the auxiliary network performance, as expected. Choosing a small sampling ratio of $\alpha = 0.25$ on the other hand is expected to cause a significant loss of information including necessary feature information. The same is evident in the reported results.

Table 4.8: Effect of varying sampling ratio on classification accuracies of ID and OOD Data.

α	MNIST		CIFAR-10		MIO-TCD	
	ID	OOD	ID	OOD	ID	OOD
$\alpha = 1.0$	98.6%	63.2%	97.7%	52.9%	99.9%	23.1%
$\alpha = 0.25$	76.6%	72.5%	74.8%	75.3%	69.3%	58.4%

To conclude, The work presents a novel mechanism, dubbed DIFE (Dominant Inter-level Feature Exploitation), and demonstrates it for efficient uncertainty-based OOD detection on data with the semantic shift. The presented work utilizes inter-level features extracted from a neural network-based classifier with uncertainty estimation to enhance OOD detection. The work is motivated by the shortcomings of uncertainty threshold-based OOD detection methods and aims to augment off-the-shelf UQ classifiers with an auxiliary network that is trained on the inter-level features obtained from the classification model. DIFE is shown to significantly outperform the uncertainty threshold-based method for OOD detection while keeping the ID classification accuracy constant. It exploits the intuition that distinct information in features is lost as a sample progresses through different layers

of a classification model and aims to exploit the features extracted in earlier layers i.e., layers closer to the input layer. Such layers are posed to contain important and distinct features for dissimilar samples, even if they are classified under the same label. It further enables dominant feature sampling from inter-level layers that allow for differentiating ID and OOD samples. While the novel mechanism is currently proposed for a specific application, it can be extended to different domains and has the potential to cater to a wide range of applications by adjusting the grouping criteria, including generalization on OOD data with covariate shift. In addition, a layer-specific information loss on the extracted inter-level features and corresponding overlap can be studied to identify optimal model architectures and define the complexity of data. The mechanism of DIFE is presented as a general concept with the aim to encourage the research community to look deeper into the features extracted by DL models, to achieve more robust systems.

4.3 Distance-based Out-of-Distribution Detection

In this section, a statistical approach is used to analyze the activations of a pre-trained neural network. Following the same intuition described in the previous section, distinct information can be extracted from intermediate layers and be used to distinguish between ID and OOD samples. OOD samples are likely to have different statistical properties compared to ID samples, which can be present at the early layers of the network. To compare the activation maps of different samples, a statistical distance metric is used. This metric measures the distance between the activation maps of a given sample and the activation maps of the training dataset. If the distance is large, it indicates that the sample is likely to be an OOD sample. On the other hand, if the distance is small, it indicates that the sample is likely to be an ID sample. The proposed approach does not require additional training or modification of the original network architecture.

Statistical Representation of Activation Maps

To compare the statistical properties of activation values of different samples, it is necessary to represent these activations as a distribution. One common approach is to fit these activations to a Gaussian distribution that corresponds to different filters of a given layer. Figure 4.5 provides an overview of the process. For a given CNN network with layers $L_1, \ldots, L_N$, an activation map can be extracted from layer L_i with shape $w \times h \times c$, where w, h, and c are the width, height, and depth of the map. Empirical analysis has shown that the activation values can be approximated and represented

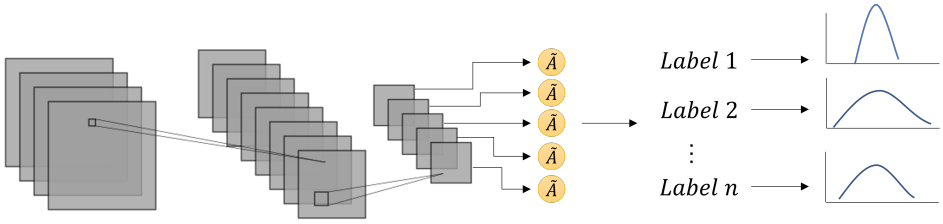


Figure 4.5: An illustration of the process of calculating the statistical distribution of activations extracted from an intermediate layer of a CNN. The class-wise distributions are used to represent each class label in the ID dataset.

by their mean value. Therefore, a single mean can be used to represent all activation values across w and h of each filter belonging to c. The mean value corresponding to an activation map $\mathcal{A}$, is computed as follows:

$$m_{lk}(x) = \frac{1}{h \cdot w} \sum_{i,j}^{h,w} \mathcal{A}(x)_{i,j} \tag{4.5}$$

For a given ID training dataset $\mathcal{D}_{ID}$ with $\mathcal{C}$ class labels, each sample is approximated using the equation 4.5. The mean and standard deviation over all samples of the class-wise groups is calculated and used to construct a distribution for each class, as follows:

$$\mu_c = \frac{1}{N_c} \sum_{i=1}^{N_c} m_{i,c} \quad \forall c \in C \tag{4.6}$$

$$\sigma_c = \sqrt{\frac{1}{N_c} \sum_{i=1}^{N_c} (m_{i,c} - \mu_c)^2} \quad \forall c \in C \tag{4.7}$$

where μ_c is the mean of class c samples, σ_c is the standard deviation, N_c is the number of samples in class c, and $m_{c,i}$ is the mean activation value of

the i-th sample in class c.

Evaluating Distributions Dissimilarities

Evaluating the statistical distance between distributions is a way to compare two distributions and determine how similar they are to each other. There are several different measures of statistical distance that can be used, depending on the specific application and the characteristics of the distributions being compared. One of the common methods is to compute the *Kullback-Leibler (KL) Divergence*, which is also known as relative entropy. KL divergence measures the amount of information lost when approximating one probability distribution with another. KL divergence is asymmetric, meaning that $\mathcal{D}_{KL}(P||Q) \neq \mathcal{D}_{KL}(Q||P)$. The KL-Divergence is calculated using the following equation:

$$\mathcal{D}_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)} \tag{4.8}$$

Here, $\mathcal{X}$ is the set of all possible values that the random variable can take, and $P(x)$ and $Q(x)$ are the probabilities of x occurring under distributions P and Q, respectively. The notation $\mathcal{D}_{KL}(P||Q)$ represents the KL divergence of P from Q.

Another measure of statistical distance is the *Jensen-Shannon divergence*, which is a symmetric version of KL divergence. The Jensen-Shannon divergence measures the similarity between two probability distributions and is calculated as the average of the KL divergences between each distribution and the average of the two distributions. the Jensen-Shannon divergence is

given by:

$$D_{JS}(P\|Q) = \frac{1}{2}\left[D_{KL}\left(P\|\frac{P+Q}{2}\right) + D_{KL}\left(Q\|\frac{P+Q}{2}\right)\right] \tag{4.9}$$

Other measures of statistical distance include the *Wasserstein distance*, also known as the Earth Mover's distance, which measures the distance between two probability distributions based on how much "work" is required to transform one distribution into the other; and the Total Variation (TV) distance, which measures the difference between the cumulative distribution functions of two probability distributions. The Wasserstein distance is given by:

$$\mathrm{W}(P,Q) = \inf_{\gamma\in\Pi(P,Q)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y)\, d\gamma(x,y) \tag{4.10}$$

where P and Q are the two probability distributions being compared, c(x,y) is the cost function for transporting a unit of mass from x to y, and γ is a joint probability distribution on X and Y whose marginals are P and Q, respectively.

In our problem formulation, the KL-Divergence will be used as the statistical distance that compares two formed distributions. The distance is utilized to decide whether the sample is an ID or OOD. The implementation details are provided in the next sections.

KL-Divergence at Inference

As explained previously, the KL divergence can be used to measure the distance between any two distributions. In the problem of OOD detection, the comparison should be made between a given sample and predetermined

distributions that correspond to the class-wise ID training dataset. To generate the ID distributions, the following steps are followed:

1. Use a subset of the ID dataset that contains enough samples and labels to create the class-wise distributions.

2. Use the ID subset and perform a forward pass. Extract class-wise intermediate features from a selected layer of the network.

3. Apply Equation 4.5 to represent each filter by its mean values.

4. Apply Equations 4.6 and 4.7 to calculate the mean and standard deviation of class-wise distributions.

Once the class-wise distributions are generated, these distributions are to be compared with another distribution generated from the testing sample, to determine if the sample is an ID or OOD. The sample, however, is a point estimate and cannot be directly used to compute the KL divergence. To overcome this challenge, it is possible to artificially create perturbations of each sample to form a distribution. One way to achieve the perturbations is by adding random Gaussian noise to each pixel of the input image, given by:

$$\tilde{x} = x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{4.11}$$

where $\tilde{x}$ is the perturbed image, x is the original input image, ϵ is the noise added to x, and σ^2 is the variance of the Gaussian distribution controlling the magnitude of the noise added. To avoid misleading the network

and to ensure the consistent classification of most perturbed samples, it is essential to use low noise variances during the perturbation process.

For each input sample, a predefined number of perturbed versions are created and a forward pass is carried out to obtain the intermediate activations from a desired middle layer. The distribution of the test sample is then obtained by following the same calculation process introduced earlier. Based on the network's classification, the two matched distributions are compared through the KL divergence. A low divergence value indicates that both distributions are similar, and therefore the test sample belongs to the ID group. Conversely, if the divergence value is high, the sample is classified as OOD. Figure 4.6 depicts a full schematic of the overall process.

Performance Evaluation

The validate the performance of the proposed mechanism is carried out by performing an OOD detection experiment. The ID dataset that is used to train the network is the MNIST dataset, while the OOD dataset is divided into the near-OOD dataset represented by the Fashion-MNIST, and the far-OOD dataset represented by CIFAR-10. Table 4.9 shows the accuracy of the proposed distance-based framework in differentiating ID samples and OOD samples. The reported results indicate that the proposed framework is capable of efficiently distinguishing between ID and OOD samples, and motivates further investigation of a broader set of benchmarking experiments.

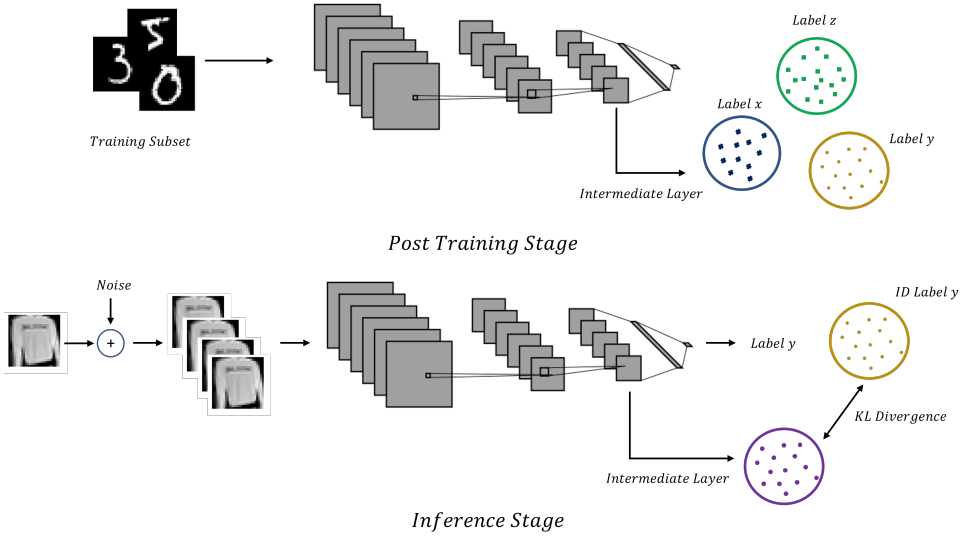


Figure 4.6: Overview of the proposed method. The upper part illustrates the process of obtaining class-wise distributions using a subset of the training data. The lower part demonstrates the inference stage, where a test sample is perturbed and then passed to the network, and the KL divergence is calculated between the sample distribution and the matched ID distribution based on the generated label.

Table 4.9: Evaluation of the distance-based OOD detection framework.

ID	OOD	Acc_{ID}	Acc_{OOD}
MNIST	Near-OOD	98.2	97.8
	Far-OOD		99.7

Chapter 5

Conclusions

5.1 Summary

This thesis presents a novel mechanism of DIFE (Dominant Inter-level Feature Exploitation) as a concept to detect OOD data based on a semantic shift. The proposed work illustrates that useful information can be extracted from intermediate layers of a DL CNN classifier (referred to it here as the primary network), and used to detect OOD data. During the classification process, images are transformed from low-level pixel values at the input side, to high-level features at intermediate layers, and finally to a single label at the output layer. DIFE exploits the intuition that distinct information in features is lost as a sample progresses through different layers of a classification model to the final layer. The mechanism aims to exploit the features extracted in earlier layers i.e., layers closer to the input layer. Such layers are posed to contain important and distinct features for dissimilar samples, even if they are classified under the same label. During training, the activations of different layers are tuned and adjusted to minimize a specific cost function. Once the training is over, the neurons of the classwise ID samples are similarly activated. As OOD samples contain dissimilar features to ID, their activations at an intermediate layer should be different. DIFE trains

a simple, yet effective auxiliary network to learn distinct features extracted from specific layers of a trained CNN classifier. Both networks run simultaneously during inference and improve the ability of the primary classifier network to detect the OOD samples. A notable feature of DIFE is that it does not require any tuning or training with OOD data. It addresses the assumption that in real-life cases, OOD samples are neither defined nor accessible, hence they cannot be used to train the network. Classifiers are hence encouraged to learn representations of ID samples and generalize to distinguish OOD samples.

While DIFE is presented as a general concept for any CNN-based classifiers (Chapter 4), the work is motivated by the shortcomings of uncertainty threshold-based OOD detection methods. UQ methods provide a well-defined framework to represent the predictive uncertainties of CNN models. Intuitively, high uncertainty outputs would indicate unfamiliar samples, hence setting a proper threshold value would reject all OODs, and allow robust deployment. An extensive analysis of the behavior of the UQ method to determine its efficiency for detecting OOD samples is presented in Chapter 3. The analysis was built on a complex, real-world dataset that contains both natural and imperfect ID samples. Additionally, to further challenge the detection task, the OOD samples were chosen to be visually similar to the ID samples. The similarity was further quantified through a statistical similarity measure calculated based on the network interpretation of the extracted features. This qualitatively validates the intuition that in realistic settings, ID and OOD samples are often very similar, and thus, harder to detect. Unfortunately, the study shows that the uncertainty threshold is

unreliable for OOD detection, especially in real-world cases, where OOD samples are semantically similar to ID samples. Augmenting DIFE with an off-the-shelf UQ-based classifier highlights the framework's ability to distinguish between the ID and OOD samples and significantly outperforms uncertainty threshold-based OOD detection. To illustrate that, a wide range of experimental evaluations have been conducted on several datasets, with different levels of correlations between IDs and OODs.

The DIFE framework includes adjusting a few parameters that specify the key elements of the approach. The sensitivity of changing these parameters has been observed and reported. The parameters define the intermediate layer selected for feature extraction and the sampling ratio of the discriminant features used to train the auxiliary network. It is essential to mention that the setting of the aforementioned parameters depends on the architecture of the network and the number of hidden layers it contains. Additionally, it depends on the complexity of the dataset being studied. This procedure results in the limitation of the proposed concept.

5.2 Future Work

This section discusses the potential expansions to the presented work. The proposed suggestions discuss both conceptual enhancements to the DIFE mechanism and possible extensions to the application field.

5.2.1 Conceptual enhancements of DIFE

The mechanism of DIFE suggests that instead of using the last layer of a CNN classifier to detect OOD samples, it is possible to examine the activations of the preceding hidden layers. The features extracted by the convolutional layers close to the input are comparable to generic color blobs while getting increasingly distinct and classwise for the following layers. In the presented work, the selection of the intermediate layer follows a heuristic approach. A future direction would be to study different methodologies that can lead to an optimal layer-specific choice for extraction. A practical way would be to incorporate a statistical distance measure between the distribution of classwise samples, and the OOD input sample. The distance would theoretically reach its greatest at a certain intermediate layer before collapsing when reaching layers close to the label side. Another consideration would be to monitor the activation of multiple intermediate hidden layers rather than a single hidden layer. The proposed modification could enhance the performance of OOD detection but with an additional computation cost and increased runtime.

DIFE employs two CNN network that operates simultaneously to detect OOD data. The objective of the first network, refer to as the primary network, is to handle the classification task and to use the generated label to define a classwise indicator vector of each ID sample. The second auxiliary network, on the other hand, is used to decide if the given sample is an ID or OOD by comparing its indicator vector to the corresponding indicator vectors of the ID classes. A future direction would be to investigate the possibility of integrating both the primary and the auxiliary networks into

one shared network that does both tasks in an end-to-end manner. The resulting single network would optimize the operation of DIFE and would replace the role of the currently proposed decision-making process, which compares the output labels of the two networks to determine if the sample is an OOD.

5.2.2 Extending the application of DIFE

The concept of DIFE was illustrated and applied to the application of semantic OOD detection in image classification. The following ideas extend DIFE to other possible applications.

DIFE with covariate shift OOD

The fundamental concept behind DIFE is its capacity to capture dominant attributes of ID data and exploit them to detect dissimilar data. In chapter 4, the effectiveness of DIFE was demonstrated with semantic shift OOD Data, i.e. new unseen samples with labels different from those in training. A prospective research area would be to examine whether DIFE can assist networks to generalize on OOD Data with covariate shifts. Ideally, the indicative vector extracted from hidden layers would determine the dominant features of classwise samples. This would aid the network to recognize samples drawn from datasets of different domains, and hence enhance the generalization of the network.

Object detection with OOD samples

Recently, there has been a surge of interest in the problem of OOD detection among the research community. Most implementations, however, are validated on image classification tasks. Determining the efficacy of OOD detection techniques on object detection is an intriguing research area. Compared to image classification, object detection is applied to more practical tasks such as autonomous driving. Therefore, it is crucial to address the untapped research potential in the area.

Dataset Expansion

Utilizing the network's capability to detect OOD classes, it would be advantageous to incorporate the identified samples for learning new class labels. This approach enables networks to generalize and adapt to previously unknown classes, thereby improving overall performance. Leveraging the concept of transfer learning, models can effectively incorporate the newly labeled data with minimal retraining procedures

Bibliography

[1] J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran, "Deep learning sensor fusion for autonomous vehicle perception and localization: A review," *Sensors*, vol. 20, no. 15, p. 4220, 2020. → pages vi, 34

[2] J. Roch, J. Fayyad, and H. Najjaran, "Dopeslam: High-precision ros-based semantic 3d slam in a dynamic environment," *Sensors*, vol. 23, no. 9, p. 4364, 2023. → pages vii

[3] W. D. Heaven, "Why meta's latest large language model survived only three days online," Nov 2022. [Online]. Available: https://www.technologyreview.com/2022/11/18/1063487/meta-large-language-model-ai-only-survived-three-days-gpt-3-science/ → pages 2

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. → pages 3, 14

[5] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330. → pages 5

[6] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017. → pages 6, 27

[7] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059. → pages 6, 25

[8] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," *Advances in Neural Information Processing Systems*, vol. 31, 2018. → pages 6, 30, 43, 78

[9] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943. → pages 10

[10] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy *et al.*, "Deepid-net: Deformable deep convolutional neural networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2403–2412. → pages 10

[11] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros *et al.*, "Development and validation of a deep learning

algorithm for detection of diabetic retinopathy in retinal fundus photographs," *Jama*, vol. 316, no. 22, pp. 2402–2410, 2016. → pages 10

[12] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2147–2156. → pages 10

[13] S. Yan, Y. Teng, J. S. Smith, and B. Zhang, "Driver behavior recognition based on deep convolutional neural networks," in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. IEEE, 2016, pp. 636–641. → pages 10

[14] Y. Zhao, J. Li, and L. Yu, "A deep learning ensemble approach for crude oil price forecasting," *Energy Economics*, vol. 66, pp. 9–16, 2017. → pages 10

[15] M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda, "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural Networks*, vol. 16, no. 5-6, pp. 555–559, 2003. → pages 11

[16] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object classification using cnn-based fusion of vision and lidar in autonomous vehicle environment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4224–4231, 2018. → pages 11

[17] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep

captioning with multimodal recurrent neural networks (m-rnn)," *arXiv preprint arXiv:1412.6632*, 2014. → pages 11

[18] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—a novel pooling deep rnn," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5271–5280, 2017. → pages 11

[19] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," *arXiv preprint arXiv:1606.01781*, 2016. → pages 11

[20] A.-r. Mohamed, G. Dahl, G. Hinton *et al.*, "Deep belief networks for phone recognition," in *Nips workshop on deep learning for speech recognition and related applications*, vol. 1, no. 9, 2009, p. 39. → pages 11

[21] C. Hongliang and Q. Xiaona, "The video recommendation system based on dbn," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. IEEE, 2015, pp. 1016–1021. → pages 11

[22] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006. → pages 11

[23] A. Krizhevsky and G. E. Hinton, "Using very deep autoencoders for

content-based image retrieval." in *ESANN*, vol. 1. Citeseer, 2011, p. 2. → pages 11

[24] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder." in *Interspeech*, vol. 2013, 2013, pp. 436–440. → pages 11

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. → pages 13, 14

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015. → pages 13

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. → pages 14

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9. → pages 14

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. → pages 15

[30] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021. → pages 16, 26

[31] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems*, vol. 30, 2017. → pages 17

[32] B. V. Dasarathy, "Sensor fusion potential exploitation-innovative architectures and illustrative applications," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 24–38, 1997. → pages 18

[33] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2020. → pages 19

[34] A. Malviya and S. Bhirud, "Wavelet based multi-focus image fusion," in *2009 Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS)*. IEEE, 2009, pp. 1–6. → pages 19

[35] D. Guan, Y. Cao, J. Yang, Y. Cao, and M. Y. Yang, "Fusion of multispectral data through illumination-aware deep neural networks for

pedestrian detection," *Information Fusion*, vol. 50, pp. 148–157, 2019. → pages 19

[36] D. Gruyer, V. Magnier, K. Hamdi, L. Claussmann, O. Orfila, and A. Rakotonirainy, "Perception, information processing and modeling: Critical stages for autonomous driving applications," *Annual Reviews in Control*, vol. 44, pp. 323–341, 2017. → pages 20

[37] D. Aubert, V. Boucher, R. Bremond, P. Charbonnier, A. Cord, E. Dumont, P. Foucher, F. Fournela, F. Greffier, D. Gruyer *et al.*, "Digital imaging for assessing and improving highway visibility," *Proceedings of the Transport Research Arena*, 2014. → pages 21

[38] A. Cord and N. Gimonet, "Detecting unfocused raindrops: In-vehicle multipurpose cameras," *IEEE Robotics & Automation Magazine*, vol. 21, no. 1, pp. 49–56, 2014. → pages 21

[39] X. Hu, F. S. A. Rodriguez, and A. Gepperth, "A multi-modal system for road detection and segmentation," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 1365–1370. → pages 21

[40] E.-J. Choi and D.-J. Park, "Human detection using image fusion of thermal and visible image with new joint bilateral filter," in *5th International Conference on Computer Sciences and Convergence Information Technology*. IEEE, 2010, pp. 882–885. → pages 21

[41] H. Torresan, B. Turgeon, C. Ibarra-Castanedo, P. Hebert, and X. P. Maldague, "Advanced surveillance systems: combining video and ther-

mal imagery for pedestrian detection," in *Thermosense XXVI*, vol. 5405. SPIE, 2004, pp. 506–515. → pages 21

[42] O. Mees, A. Eitel, and W. Burgard, "Choosing smartly: Adaptive multimodal fusion for object detection in changing environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 151–156. → pages 21

[43] M. Vandersteegen, K. V. Beeck, and T. Goedemé, "Real-time multispectral pedestrian detection with a single-pass deep neural network," in *International Conference Image Analysis and Recognition*. Springer, 2018, pp. 419–426. → pages 21

[44] P. Fritsche, B. Zeise, P. Hemme, and B. Wagner, "Fusion of radar, lidar and thermal information for hazard detection in low visibility environments," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, 2017, pp. 96–101. → pages 21

[45] T. Salimans, D. Kingma, and M. Welling, "Markov chain monte carlo and variational inference: Bridging the gap," in *International conference on machine learning*. PMLR, 2015, pp. 1218–1226. → pages 25

[46] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118. → pages 25

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks

from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014. → pages 25

[48] M. Combalia, F. Hueto, S. Puig, J. Malvehy, and V. Vilaplana, "Uncertainty estimation in deep neural networks for dermoscopic image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 744–745. → pages 26

[49] M.-H. Laves, S. Ihler, J. F. Fast, L. A. Kahrs, and T. Ortmaier, "Well-calibrated regression uncertainty in medical imaging with deep learning," in *Medical Imaging with Deep Learning*. PMLR, 2020, pp. 393–412. → pages 26

[50] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks," *Neurocomputing*, vol. 338, pp. 34–45, 2019. → pages 26

[51] S. Boluki, R. Ardywibowo, S. Z. Dadaneh, M. Zhou, and X. Qian, "Learnable bernoulli dropout for bayesian deep learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 3905–3916. → pages 26

[52] B. Phan, R. Salay, K. Czarnecki, V. Abdelzad, T. Denouden, and S. Vernekar, "Calibrating uncertainties in object localization task," *arXiv preprint arXiv:1811.11210*, 2018. → pages 26

[53] Y. Cao, T. A. Geddes, J. Y. H. Yang, and P. Yang, "Ensemble deep

learning in bioinformatics," *Nature Machine Intelligence*, vol. 2, no. 9, pp. 500–508, 2020. → pages 27

[54] L. Nanni, S. Ghidoni, and S. Brahnam, "Ensemble of convolutional neural networks for bioimage classification," *Applied Computing and Informatics*, 2020. → pages 27

[55] X. Dai, X. Wu, B. Wang, and L. Zhang, "Semisupervised scene classification for remote sensing images: A method based on convolutional neural networks and ensemble learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 6, pp. 869–873, 2019. → pages 27

[56] F. Lv, M. Han, and T. Qiu, "Remote sensing image classification based on ensemble extreme learning machine with stacked autoencoder," *IEEE Access*, vol. 5, pp. 9021–9031, 2017. → pages 27

[57] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016. → pages 27

[58] H. Wang, Y. Yu, Y. Cai, X. Chen, L. Chen, and Y. Li, "Soft-weighted-average ensemble vehicle detection method based on single-stage and two-stage deep learning models," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 1, pp. 100–109, 2020. → pages 27

[59] N. Mahendran, D. R. Vincent, K. Srinivasan, C.-Y. Chang, A. Garg, L. Gao, and D. G. Reina, "Sensor-assisted weighted average ensemble model for detecting major depressive disorder," *Sensors*, vol. 19, no. 22, p. 4822, 2019. → pages 27

[60] F. Divina, A. Gilson, F. Goméz-Vela, M. García Torres, and J. F. Torres, "Stacking ensemble learning for short-term electricity consumption forecasting," *Energies*, vol. 11, no. 4, p. 949, 2018. → pages 27

[61] Y. Wang, D. Wang, N. Geng, Y. Wang, Y. Yin, and Y. Jin, "Stacking-based ensemble learning of decision trees for interpretable prostate cancer detection," *Applied Soft Computing*, vol. 77, pp. 188–204, 2019. → pages 27

[62] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher *et al.*, "A survey of uncertainty in deep neural networks," *arXiv preprint arXiv:2107.03342*, 2021. → pages 27

[63] E. J. Herron, S. R. Young, and T. E. Potok, "Ensembles of networks produced from neural architecture search," in *International Conference on High Performance Computing*. Springer, 2020, pp. 223–234. → pages 29

[64] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," *Advances in neural information processing systems*, vol. 31, 2018. → pages 30

[65] T. Joo, U. Chung, and M.-G. Seo, "Being bayesian about categorical probability," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4950–4961. → pages 30

[66] A. Amini, W. Schwarting, A. Soleimany, and D. Rus, "Deep eviden-

tial regression," *Advances in Neural Information Processing Systems*, vol. 33, pp. 14 927–14 937, 2020. → pages 30

[67] G. Parisi and R. Shankar, "Statistical field theory," 1988. → pages 30

[68] M. Itkina and M. J. Kochenderfer, "Interpretable self-aware neural networks for robust trajectory prediction," *arXiv preprint arXiv:2211.08701*, 2022. → pages 30

[69] K. Zou, X. Yuan, X. Shen, M. Wang, and H. Fu, "Tbrats: Trusted brain tumor segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2022, pp. 503–513. → pages 32

[70] Z. Liu, A. Amini, S. Zhu, S. Karaman, S. Han, and D. L. Rus, "Efficient and robust lidar-based end-to-end navigation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 247–13 254. → pages 32

[71] D. Feng, Y. Cao, L. Rosenbaum, F. Timm, and K. Dietmayer, "Leveraging uncertainties for deep multi-modal object detection in autonomous driving," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 877–884. → pages 32

[72] C. Wang, X. Wang, J. Zhang, L. Zhang, X. Bai, X. Ning, J. Zhou, and E. Hancock, "Uncertainty estimation for stereo matching based on evidential deep learning," *Pattern Recognition*, vol. 124, p. 108498, 2022. → pages 33

[73] J. Zhang, Y. Chen, and Z. Tu, "Uncertainty-aware 3d human pose estimation from monocular video," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 5102–5113. → pages 33

[74] T. Zhu, K. Li, P. Herrero, and P. Georgiou, "Personalized blood glucose prediction for type 1 diabetes using evidential deep learning and meta-learning," *IEEE Transactions on Biomedical Engineering*, 2022. → pages 33

[75] S. Faghani, B. Khosravi, K. Zhang, M. Moassefi, J. M. Jagtap, F. Nugen, S. Vahdati, S. P. Kuanar, S. M. Rassoulinejad-Mousavi, Y. Singh *et al.*, "Mitigating bias in radiology machine learning: 3. performance metrics," *Radiology: Artificial Intelligence*, vol. 4, no. 5, p. e220061, 2022. → pages 33

[76] A. P. Soleimany, A. Amini, S. Goldman, D. Rus, S. N. Bhatia, and C. W. Coley, "Evidential deep learning for guided molecular property prediction and discovery," *ACS central science*, vol. 7, no. 8, pp. 1356–1367, 2021. → pages 33

[77] R. Wason, "Deep learning: Evolution and expansion," *Cognitive Systems Research*, vol. 52, pp. 701–708, 2018. → pages 33

[78] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Machine Learning*, vol. 110, no. 3, pp. 457–506, 2021. → pages 34, 41

[79] J. Guo, U. Kurup, and M. Shah, "Is it safe to drive? an overview of factors, metrics, and datasets for driveability assessment in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3135–3151, 2019. → pages 34

[80] M. Paschali, S. Conjeti, F. Navarro, and N. Navab, "Generalizability vs. robustness: investigating medical imaging networks using adversarial examples," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 493–501. → pages 34

[81] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," *arXiv preprint arXiv:2110.11334*, 2021. → pages 34

[82] K. Lee, K. Lee, H. Lee, and J. Shin, "A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018. → pages 35

[83] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," 2016, pp. 2414–2423. → pages 35

[84] V. Cheplygina, M. de Bruijne, and J. P. Pluim, "Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis," *Medical image analysis*, vol. 54, pp. 280–296, 2019. → pages 35

[85] Z. Liu, Z. Miao, X. Pan, X. Zhan, D. Lin, S. X. Yu, and

B. Gong, "Open compound domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 406–12 415. → pages 35

[86] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. → pages 35

[87] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 4, pp. 801–814, 2018. → pages 35

[88] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4893–4902. → pages 35

[89] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, "Learning to Generate Novel Domains for Domain Generalization," in *Computer Vision – ECCV 2020*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 561–578. → pages 35

[90] H. Li, S. J. Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5400–5409. → pages 35

[91] L. Tran, K. Sohn, X. Yu, X. Liu, and M. Chandraker, "Gotta adapt'em

all: Joint pixel and feature-level domain adaptation for recognition in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2672–2681. → pages 35

[92] X. Zhang, F. X. Yu, S.-F. Chang, and S. Wang, "Deep transfer network: Unsupervised domain adaptation," *arXiv preprint arXiv:1503.00591*, 2015. → pages 35

[93] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *ICML*, 2011. → pages 35

[94] R. Volpi and V. Murino, "Addressing model vulnerability to distributional shifts over image transformation sets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7980–7989. → pages 36

[95] Z. Xu, D. Liu, J. Yang, C. Raffel, and M. Niethammer, "Robust and generalizable visual representation learning via random convolutions," *arXiv preprint arXiv:2007.13003*, 2020. → pages 36

[96] S. Seo, Y. Suh, D. Kim, G. Kim, J. Han, and B. Han, "Learning to optimize domain specific normalization for domain generalization," in *European Conference on Computer Vision*. Springer, 2020, pp. 68–83. → pages 36

[97] Y. Li, Y. Yang, W. Zhou, and T. Hospedales, "Feature-critic networks for heterogeneous domain generalization," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3915–3924. → pages 36

[98] C. Geng, S.-J. Huang, and S. Chen, "Recent Advances in Open Set Recognition: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3614–3631, Oct. 2021, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. → pages 38

[99] J. Tack, S. Mo, J. Jeong, and J. Shin, "CSI: Novelty Detection via Contrastive Learning on Distributionally Shifted Instances," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 11 839–11 852. → pages 38

[100] D. Hendrycks and K. Gimpel, "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks," Oct. 2018. → pages 38, 71

[101] S. Liang, Y. Li, and R. Srikant, "Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks," Aug. 2020. → pages 38

[102] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira, "Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 951–10 960. → pages 40

[103] D. Macêdo, T. I. Ren, C. Zanchettin, A. a. L. Oliveira, and T. Ludermir, "Entropic out-of-distribution detection: Seamless detection of unknown examples," *IEEE Transactions on Neural Networks and Learning Systems*, 2021. → pages 40

[104] D. Macêdo and T. Ludermir, "Enhanced isotropy maximization loss: Seamless and high-performance out-of-distribution detection simply replacing the softmax loss," *arXiv preprint arXiv:2105.14399*, 2021. → pages 40

[105] D. Macêdo, C. Zanchettin, and T. Ludermir, "Distinction maximization loss: Efficiently improving out-of-distribution detection and uncertainty estimation by replacing the loss and calibrating," *arXiv preprint arXiv:2205.05874*, 2022. → pages 40

[106] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017. → pages 41

[107] A. A. M. Al-Saffar, H. Tao, and M. A. Talab, "Review of deep convolution neural network in image classification," in *2017 International conference on radar, antenna, microwave, electronics, and telecommunications (ICRAMET)*. IEEE, 2017, pp. 26–31. → pages 41

[108] J. Caldeira and B. Nord, "Deeply uncertain: comparing methods of uncertainty quantification in deep learning algorithms," *Machine Learning: Science and Technology*, vol. 2, no. 1, p. 015002, 2020. → pages 41

[109] D. Ulmer and G. Cinà, "Know your limits: Monotonicity & softmax make neural classifiers overconfident on ood data," *arXiv preprint arXiv:2012.05329*, 2020. → pages 42

[110] K. Stacke, G. Eilertsen, J. Unger, and C. Lundström, "Measuring

domain shift for deep learning in histopathology," *IEEE journal of biomedical and health informatics*, vol. 25, no. 2, pp. 325–336, 2020. → pages 46

[111] Z. Luo, F. Branchaud-Charron, C. Lemaire, J. Konrad, S. Li, A. Mishra, A. Achkar, J. Eichel, and P.-M. Jodoin, "MIO-TCD: A New Benchmark Dataset for Vehicle Classification and Localization," *IEEE Transactions on Image Processing*, vol. 27, no. 10, Oct. 2018, conference Name: IEEE Transactions on Image Processing. → pages 49, 69, 77

[112] G. Fei and B. Liu, "Breaking the closed world assumption in text classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 506–514. [Online]. Available: https://aclanthology.org/N16-1061 → pages 62

[113] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., 2014. → pages 64

[114] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012. → pages 69, 77

[115] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009. → pages 69, 77

[116] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017. → pages 69

[117] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008. → pages 69

[118] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011. → pages 69

[119] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *arXiv preprint arXiv:1706.02690*, 2017. → pages 71

[120] D. Hendrycks, S. Basart, M. Mazeika, M. Mostajabi, J. Steinhardt, and D. Song, "Scaling out-of-distribution detection for real-world settings," *arXiv preprint arXiv:1911.11132*, 2019. → pages 71

[121] A. Schwaiger, P. Sinhamahapatra, J. Gansloser, and K. Roscher, "Is uncertainty quantification in deep learning sufficient for out-of-distribution detection?" in *AISafety@ IJCAI*, 2020. → pages 78