A study on open set recognition methods

Sun, Xin

2021

Sun, X. (2021). A study on open set recognition methods. Doctoral thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/152461>

<https://hdl.handle.net/10356/152461>

<https://doi.org/10.32657/10356/152461>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 19:34:54 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

A Study on Open Set Recognition Methods

SUN XIN

School of Electrical & Electronic Engineering

2021

A Study on Open Set Recognition Methods

SUN XIN

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

10-02-2021

.....

Date

sun xin

.....

SUN XIN

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

10-02-2021

.....

Date

.....

Ling Keck-Voon

Authorship Attribution Statement

The thesis contains material from three papers published in or submitted to the following peer-reviewed journals or conference proceedings where I was the first author.

Chapter 3 is accepted by:

X. Sun, K. V. Ling, K. K. Sin and Y. Liu. *Open Set Recognition Method for Air Leakage Detection of Pneumatic Train Door Subsystem.* IEEE Transactions on Instrumentation and Measurement, Paper No. TIM-21-00333.

The contributions of the co-authors are as follows:

- Assoc/Prof Ling Keck-Voon supervised me on the overall research direction and edited the manuscript drafts.
- I studied on the algorithm, carried out the experiments and prepared the manuscript drafts.
- Mr. Sin Kok-Kee built up the train door test jig, installed the embedded controller and the pressure transducer, and collected air pressure data.
- Mr. Liu Yang provided valuable comments on the manuscript.

Chapter 4 is published as:

X. Sun, Z. Yang, C. Zhang, K. V. Ling and G. Peng. *Conditional gaussian distribution learning for open set recognition.* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020 (pp. 13480-13489);

The contributions of the co-authors are as follows:

- Assoc/Prof Ling Keck-Voon supervised me on the overall research direction and edited the manuscript drafts.
- I studied on the algorithm, carried out the experiments and prepared the manuscript drafts.
- Mr. Yang Zhenning provided the codes of the unknown detector.
- Mr. Zhang Chi helped to analyze results and provided valuable comments on the manuscript.
- Mr. Peng Guohao developed the algorithm and provided valuable comments on the manuscript.

Chapter 5 is submitted to:

X. Sun, H. Ding, C. Zhang, G. Lin and K. V. Ling. *M2IOSR: Maximum Mutual Information Open Set Recognition*. IEEE Transactions on Image Processing (TIP-25736-2021).

The contributions of the co-authors are as follows:

- Assoc/Prof Ling Keck-Voon and Asst/Prof Lin Guoshen supervised me on the overall research direction and edited the manuscript drafts.
- I studied on the algorithm, carried out the experiments and prepared the manuscript drafts.
- Dr. Ding Henghui and Mr. Zhang Chi helped to analyze results and provided valuable comments on the manuscript.

10-02-2021

.....

Date

sun xin

.....

SUN XIN

Acknowledgements

First and foremost, I wish to express my greatest gratitude to my supervisor, Professor Ling Keck-Voon. He constantly encouraged and supported me throughout my Ph.D. career. He has helped me a lot in how to do academic research, in-depth thinking, and manuscript writing. I would like to thank Professor Lin Guoshen for his patient guidance and for leading me into the wonderful world of deep learning. Also, I would like to thank Nanyang Technological University, National Research Foundation Singapore, and SMRT Corporation for funding my education and research.

My sincere thanks also go to my colleagues (Dr. Sun Quqin, Dr. Ding Henghui, Dr. Liao Xinqin, Dr. Li Kangrong, Dr. Li Tianliang, Dr. Pritom Jyoti Bora, Dr. Li Yuhua, Dr. Xiao Jing, Dr. Chua Eng-Kee, Mr. Sin Kok-Kee, Mr. Zhao Zhenyu, Mr. Yang Zhenning, Mr. Liu Yang, Mr. Zhang Chi, Mr. Zhang Zhixiang, Mr. Peng Guohao, and Mr. Kalyankar Sharvan Kumar) for their warm-hearted assistance during my Ph.D. career.

Finally, I wish to convey my love and heartfelt gratitude to my parents, my family, and my girlfriend, who have generously offered me love, support, and encouragements.

Abstract

Deep learning has achieved state-of-the-art performance in many computer vision tasks, but there are still multiple challenges when applying deep learning to real-world problems. One typical challenge is that incomplete knowledge exists during the training phase, and an unknown sample may be fed into the system during the testing phase, then the network will wrongly recognize it as one of the known classes. A potential solution to address this issue is Open Set Recognition (OSR), which assumes that deficient knowledge exists during the training stage, and aims to not only classify samples of known classes but detect unknown samples during the testing stage.

OSR methods are gaining more and more attention these years. Considering that information can only be extracted from known samples, to realize unknown detection, some OSR methods are based on generative models (such as Generative Adversarial Networks (GAN) or Variational Auto-Encoders (VAE)). On the other hand, some OSR methods synthesize unknown data or redistribute the prediction scores among known classes to directly calculate out prediction scores of unknown samples. Although these methods achieve state-of-the-art performance in computer vision tasks, most of them have a long running time and are not applicable to real-time industrial tasks. To have a shorter running time, we proposed an OSR method, called Discriminative Loss. We combine the proposed loss function with the Softmax loss function, which is used in most Convolutional Neural Networks (CNNs), to force learned features in different classes to be close to different centroids for Gaussian modeling. The proposed method is demonstrated on the MNIST dataset. This method is also applied to detect air leakage on pneumatic train door subsystems and achieves promising classification accuracy with a shorter running time.

Although the proposed Discriminative Loss method can be successfully applied to industrial sensor data, its performance is not satisfying on more complex data (*i.e.*, image). To achieve higher accuracy on image datasets, we propose a novel OSR method, called Conditional Probabilistic Generative Models (CPGM). The core insight of this method

is to add discriminative information into the probabilistic generative models (*i.e.*, Variational Auto-encoders (VAE) and Adversarial Auto-encoders (AAE)), such that the proposed models cannot only detect unknown samples but also classify known classes by forcing different latent features to approximate class conditional Gaussian distributions.

Auto-encoders (AE) is widely used to generate latent features with desired statistical characteristics for OSR. Meanwhile, AE also can extract class-specific features through its reconstruction training strategy. This reconstruction strategy requires the network to restore the input image on pixel-level. However, this strategy is commonly over-demanding for OSR since class-specific features are generally contained in target objects, not in all pixels. To address this shortcoming, we propose a mutual information-based method with a streamlined architecture, Maximal Mutual Information Open Set Recognition (M2IOSR). The proposed M2IOSR only uses an encoder to extract class-specific features by maximizing the mutual information between the given input and its latent features across multiple scales. Meanwhile, to further reduce the open space risk, latent features are constrained to class conditional Gaussian distributions by a KL-divergence loss function.

We compare the proposed three methods on six standard image datasets (MNIST, SVHN, CIFAR-10, CIFAR-100, ImageNet, and LSUN) and the air pressure data collected from a pneumatic train door subsystem. All the proposed methods do not significantly decline closed set classification accuracies while realizing OSR. For experiments on standard image datasets, CPGM significantly outperforms the baseline methods, and M2IOSR achieves state-of-the-art performance. For experiments on air pressure data, Gaussian-based models (Discriminative Loss, CPGM-VAE, and CPGM-AAE) all achieve promising results. Among them, CPGM-AAE achieves the highest F_1 scores in both the extension phase and the retraction phase. It is also worth noting that compared with other OSR methods, Discriminative Loss has a minimal number of parameters, and its running time is only 0.2 milliseconds per input slower than that of the fastest method (Softmax).

Contents

Acknowledgements	xi
Abstract	xiii
List of Figures	xix
List of Tables	xxiii
Symbols and Acronyms	xxv
1 Introduction	1
1.1 Background	1
1.2 Challenges and Motivations	2
1.3 Proposed OSR methods	4
1.4 Contributions	6
1.5 Outline of the Thesis	7
2 Literature Review	9
2.1 Deep Learning Models	9
2.1.1 VGG Neural Networks	9
2.1.2 Variational Auto-encoders	10
2.1.3 Adversarial Auto-encoders	12
2.2 Open Set Recognition	13
2.3 Unknown Detection	14
2.4 Mutual Information for Unsupervised Learning	15
3 Open Set Recognition with Short Running Time	17
3.1 Introduction	18
3.2 Methodology	19
3.3 The Pneumatic Train Door Subsystem	23
3.3.1 Pneumatic Module	24
3.3.2 Data Acquisition	26
3.3.3 Data Processing	27
3.4 Experiments and Results	28

3.4.1	Implementation Details	28
3.4.2	Air Leakage Classification	29
3.4.3	OSR on Air Pressure Data	31
3.4.4	Analysis of Efficiency	35
3.5	Chapter Summary	36
4	Open Set Recognition with Conditional Probabilistic Generative Models	37
4.1	Introduction	37
4.2	Preliminary	39
4.2.1	VAE	39
4.2.2	AAE	41
4.3	Method	41
4.3.1	CPGM-VAE	42
4.3.1.1	Architecture	42
4.3.1.2	Training	44
4.3.1.3	Testing	46
4.3.2	CPGM-AAE	46
4.3.2.1	Architecture	46
4.3.2.2	Training	49
4.3.2.3	Testing	50
4.4	Experiments	50
4.4.1	Implementation Details	50
4.4.2	Dataset Description	51
4.4.3	Ablation Study of CPGM-VAE	52
4.4.4	Model Analysis of CPGM-AAE	54
4.4.5	Comparison with State-of-the-art Results	57
4.4.6	CPGM-VAE vs. CPGM-AAE	59
4.5	Chapter Summary	61
5	Maximal Mutual Information Open Set Recognition	63
5.1	Introduction	64
5.2	Preliminaries	65
5.3	Proposed Method	66
5.3.1	Loss Function Derivation	67
5.3.2	Mutual Information Maximization	68
5.3.3	Training and Testing	71
5.4	Experiments	72
5.4.1	Implementation Details	72
5.4.2	Dataset Description	72
5.4.3	Ablation Analysis	73
5.4.4	Case Study	75
5.4.5	Comparison with State-of-the-art Results	78
5.5	Chapter Summary	81

6	Comparison of Proposed OSR Methods	83
6.1	Comparison on Image Datasets	83
6.2	Comparison on Air Pressure Data	86
6.3	Comparison of Running Time	88
7	Conclusion and Future Work	89
7.1	Conclusion	89
7.2	Future Work	91
	Author's Publications	93
	Bibliography	95

List of Figures

1.1	Comparison of latent representations on the MNIST dataset of VAE (a), AAE (b), and proposed methods: CPGM-VAE (c) and CPGM-AAE (d). VAE and AAE are widely used in unknown detection, but they cannot provide discriminative features to undertake classification tasks as all features just follow one distribution. Comparatively, the proposed method can learn conditional Gaussian distributions by forcing different latent features to approximate different Gaussian models, which enables the proposed method to classify known samples as well as reject unknown samples.	4
1.2	Class activation maps produced by auto-encoders and the proposed method M2IOSR. Due to the reconstruction of all pixels including target objects and background stuff, auto-encoders make the features of target objects less pronounced (<i>e.g.</i> , the first two images). By contrast, the proposed method M2IOSR can extract more class-specific features, and thus its extracted features focus more on target objects. In the images where the ratio of the background is small, <i>e.g.</i> , the third image, our method localizes the target object more accurately than auto-encoders.	5
2.1	Architecture of VGG-13.	10
2.2	Framework of VAE when the prior distribution is the normal distribution.	11
2.3	Framework of AAE when the prior distribution is the normal distribution.	13
3.1	Comparison of latent features learned from a LeNet model (a CNN model) on the MNIST dataset between using the Softmax loss (a) and using the proposed Discriminative Loss (b).	20
3.2	The difference between linear classification, domain description, and discriminative features.	21
3.3	Testing accuracies using Discriminative Loss on the MNIST dataset with different λ	22
3.4	A test jig replicating the actual pneumatic train door subsystem. The actuator is marked with a green border, the pressure transducer is marked with a blue border, and the embedded controller is marked with a red border.	24
3.5	The framework of the proposed online air leakage detection system. This system could detect different levels of air leakage through the following three steps: data acquisition, signal processing, and air leakage detection based on the proposed OSR method.	25

3.6	The schematic illustration of the pneumatic module used to control the door movement. The direction of airflow when door opening is shown by the red line, while the direction of airflow when door closing is shown by the green line.	26
3.7	A typical pressure signal in one cycle, which is made up of two phases: extension (door opening) and retraction (door closing). The extension signal and retraction signal are similar but not exactly the same.	26
3.8	Air pressure signals collected in five conditions (<i>normal, leakage 1, 2, 3, and 4</i>) after mean filtering and normalization in the extension phase (a) and retraction phase (b).	27
3.9	The process of generating a signal image by the 2D transformation.	28
3.10	The architecture of the constructed CNN (C : convolutional layer; M : max-pooling layer; R : ReLU layer; F : fully-connected layer).	29
3.11	Comparison of latent features learned from the constructed CNN on the pressure training dataset between using the Softmax loss function (extension phase (a), retraction phase (b)) and using Discriminative Loss where λ is set equal to 0.5 (extension phase (c), retraction phase (d)).	30
3.12	The feature distribution learned from the trained CNN model using Discriminative Loss ($\lambda = 0.5$) on the pressure testing dataset in the extension phase (a) and retraction phase (b).	32
3.13	The Macro- F_1 and Micro- F_1 scores experimented on air pressure data with different λ in the extension phase (a) and retraction phase (b).	32
3.14	The definition of Area and Interval in the extension phase.	33
4.1	Block diagram of CPGM-VAE: The encoder $\mathcal{F}$ and decoder $\mathcal{G}$ are applied with the probabilistic ladder architecture to generate more informative posterior distributions. The known classifier $\mathcal{C}$ takes latent representations as input and produces the probability distribution over the known classes. The unknown detector $\mathcal{D}$ is modeled by the conditional Gaussian distributions and reconstruction errors from training samples, which is used for unknown detection. During training, the proposed model is trained to minimize the sum of the reconstruction loss $\mathcal{L}_r$, KL-divergence $\mathcal{L}_{KL}$ (both in the latent space and middle layers) and classification loss $\mathcal{L}_c$. During testing, the unknown detector $\mathcal{D}$ will judge whether this sampler is unknown by its latent features and reconstruction errors. If this sample is known, the known classifier $\mathcal{C}$ will give out its predicted label.	42
4.2	Operations in the upward pathway of CPGM-VAE.	43
4.3	Operations in the downward pathway of CPGM-VAE.	44

4.4	The model architecture of the proposed CPGM-AAE (a) (details of CPGM-AAE are described in Section 4.3.2.1) and two model variants (b) and (c) (details of two model variants are described in Section 4.4.4). We visualize the features of samples from known classes generated by these three models in (d) , (e) and (f) , respectively. All these three models could generate 10 feature clusters and each cluster represents one class. Although AAE-Variant 1 could generate separable features, these features do not follow class conditional Gaussian distributions for unknown detection. Conversely, AAE-Variant 2 could generate ideal class conditional Gaussian distributions, but they are highly overlapped and can hardly provide discriminative information for known classification. In comparison, the proposed method, CPGM-AAE, could generate ideal class conditional Gaussian distributions for unknown detection, and these features are also separable in the feature space for known classification.	48
4.5	F_1 -scores against varying Openness with different baselines (described in Section 4.4.3) for the ablation study of CPGM-VAE. CPGM-VAE achieves the best F_1 -scores by using conditional Gaussian distributions (CGD) and reconstruction errors (RE) jointly to detect unknown samples with the probabilistic ladder architecture.	53
4.6	F_1 -scores against varying Openness with three AAE-based models (described in Section 4.4.4). Among these three models, CPGM-AAE always achieves the best F_1 -scores under varying Openness.	56
4.7	F_1 -scores against varying Openness with different baselines (described in Section 4.4.4) for the ablation study of CPGM-AAE. CPGM-AAE achieves the best F_1 -scores by using conditional Gaussian distributions (CGD) and reconstruction errors (RE) jointly to detect unknown samples.	57
4.8	Examples from MNIST, Omniglot, MNIST-Noise, and Noise datasets.	57
4.9	tSNE [1] visualization of feature distributions of testing samples in CPGM-VAE (a) and CPGM-AAE (b) . In both methods, samples from different known classes distribute in different clusters while unknown samples are mostly located in open space. The comparison between CPGM-VAE and CPGM-AAE shows that the variance of each cluster is large in CPGM-VAE and the clusters of known classes are more compact in CPGM-AAE. Therefore, when using conditional Gaussian distributions for OSR, CPGM-AAE yields better results.	59
4.10	Reconstruction error histograms of known and unknown classes in CPGM-VAE (a) and CPGM-AAE (b) . CPGM-AAE has a more clear decision boundary between known and unknown classes, which explains why additionally using reconstruction errors to detect unknown samples at test time can significantly improve performance on CPGM-AAE (the comparison between ‘CAAE+CGD’ and ‘CPGM-AAE (proposed)’ as shown in Figure 4.7) but no obvious improvement on CPGM-VAE (the comparison between ‘LCVAE+CGD’ and ‘CPGM-VAE (proposed)’ as shown in Figure 4.5).	60

4.11	Reconstructed images generated from CPGM-AAE trained on known classes. All reconstructed images take the form of a blurry version of the input images, but details are preserved better in reconstructions of known samples. Predicting labels from reconstructions of unknown samples is difficult. This model seems to generate a ‘cat’ from a ship and generate a ‘dog’ from a truck (marked in red borders).	61
5.1	Overview architecture of the proposed method on positive pairs. $f_d(\mathbf{x})$ represents a feature with a spatial size of $d \times d$. During the training phase, we maximize global MI on $[f_{16}(\mathbf{x}), f_1(\mathbf{x})]$ to extract useful information of the input, and maximize local MI on $[f_{16}(\mathbf{x}), f_1(\mathbf{x})]$, $[f_4(\mathbf{x}), f_1(\mathbf{x})]$ and $[f_4(\mathbf{x}), f_4(\mathbf{x})]$ to filter out useless features and prevent MI from being lost in the middle layers. Meanwhile, to further reduce open space risk [2], latent features $f_1(\mathbf{x})$ are constrained to class conditional Gaussian distributions. A 2-dimensional tSNE [1] visualization of $f_1(\mathbf{x})$ distributions on the CIFAR10 dataset [3] is shown at the bottom-left. . . .	69
5.2	F1-scores against varying Openness levels in ablative experiments. The proposed method outperforms other baselines and achieves the highest F1-scores on all Openness levels.	75
5.3	Class activation maps produced by auto-encoders and the proposed method M2IOSR.	76
5.4	Histograms of maximum prediction probability on the known testing set and unknown testing set for the AE model and M2IOSR. More unknown samples have smaller prediction probabilities in the M2IOSR, making it more effective to detect unknown samples through a threshold.	77
5.5	tSNE [1] visualization of feature distributions of testing samples in M2IOSR. Samples in known classes are distributed into six clusters, while most unknown samples are located in the open space.	78
5.6	Confusion Matrix. Here we show the confusion matrices produced by the AE model and our approach. The vehicle classes (airplane, automobile, ship, and truck) form the unknown class set and are not used in the training stage. Our approach performs much better on detecting unknown samples (the bottom-right part) than the AE model and significantly reduces errors in classifying unknown samples into known categories (the bottom-left part).	79

List of Tables

1.1	Comparison between traditional classification, anomaly detection and open set recognition	2
3.1	The parameters in the constructed CNN.	29
3.2	Comparison of closed set testing accuracy on air pressure data between using the Softmax loss function and using the proposed loss function. .	30
3.3	Macro- F_1 and Micro- F_1 scores among five methods for the extension phase.	34
3.4	Macro- F_1 and Micro- F_1 scores among five methods for the retraction phase.	34
3.5	Comparison of the process in the testing phase among five methods. . .	35
3.6	Comparison of running time (milliseconds per input) among five deep learning-based OSR methods.	35
4.1	Comparison of closed set testing accuracies between the plain CNN and the proposed CPGM methods. Although the training objective of CPGM is classifying known samples as well as learning conditional Gaussian distributions, there is no significant degradation in closed set accuracy. .	51
4.2	Open set classification results on the MNIST dataset with various outliers added to the test set as unknowns. The performance is evaluated by macro-averaged F_1 -scores in 11 classes (10 known classes and <i>unknown</i>). Our methods significantly outperform the baseline methods. . .	58
4.3	Open set classification results on the CIFAR-10 dataset with various outliers added to the test set as unknowns. The performance is evaluated by macro-averaged F_1 -scores in 11 classes (10 known classes and <i>unknown</i>). Our methods achieve the best performance among baseline methods. *We report the experimental results reproduced in [4].	58
5.1	Global discriminator architecture.	70
5.2	Local discriminator architecture.	70
5.3	Comparison of testing accuracies on closed sets between vanilla CNN models and our proposed approach M2IOSR. The training objective of M2IOSR is to maximize mutual information as well as classify known classes, but there is no large decline in testing accuracies on closed sets. .	72
5.4	Averaged F1-scores in five randomized trials.	78

5.5	F1-scores on the CIFAR10 dataset with four other datasets added to the testing set as unknown.	80
5.6	F1-scores on the MNIST dataset with three other datasets added to the testing set as unknown.	80
5.7	Comparison of running time and number of parameters. All the models are based on VGG-13.	81
6.1	Comparison of closed set testing accuracies between the vanilla CNN and the proposed three methods.	84
6.2	F_1 -scores on the MNIST dataset with three other datasets added to the testing set as unknown.	85
6.3	F_1 -scores on the CIFAR-10 dataset with four other datasets added to the testing set as unknown.	86
6.4	Averaged F_1 -scores in five randomized trials.	86
6.5	Macro- F_1 and Micro- F_1 scores on the air pressure dataset for the extension phase and retraction phase.	87
6.6	Comparison of running time and number of parameters. All the models are based on VGG-13.	88

Symbols and Acronyms

Symbols

$D_{KL}(A B)$	the KL-divergence between A and B
$\mathcal{N}(\cdot)$	the normal distribution
$\odot$	the element-wise product
$\mathbb{E}[\cdot]$	the mathematical expectation
$\sum$	the summation operation
$\ \cdot\ _2$	the L2-norm operation
$\max(\cdot)$	the maximum operation
$\min(\cdot)$	the minimum operation
$I(A; B)$	the mutual information between A and B
$H(\cdot)$	the entropy operation

Acronyms

2D	Two Dimensional
AAE	Adversarial Auto-encoders
AE	Auto-encoders
AI	Artificial Intelligence
AUROC	Area Under the Receiver Operating Characteristic curve
CAM	Class Activation Mapping
CGAN	Conditional Generative Adversarial Networks
CNN	Convolutional Neural Network
CPGM	Conditional Probabilistic Generative Models
ESN	Echo-State Network
GAN	Generative Adversarial Networks

GMM	Gaussian Mixture Model
JS	Jensen–Shannon
KL	Kullback–Leibler
KPCA	Kernel Principal Component Analysis
M2IOSR	Maximal Mutual Information Open Set Recognition
MI	Mutual Information
MSVM	Multi-class Support Vector Machine
OSR	Open Set Recognition
PCA	Principal Component Analysis
PReLU	Parametric Rectified Linear Unit
RBF	Radial Basis Function
RBM	Restricted Boltzmann Machine
ReLU	Rectified Linear Unit
RGB	Red Green Blue
RMS	Root Mean Square
ROC	Receiver Operating Characteristic
STD	Standard Deviation
SVM	Support Vector Machine
VAE	Variational Auto-encoders
VGG	Visual Geometry Group

Chapter 1

Introduction

1.1 Background

Deep learning-based methods have demonstrated remarkable success in many recognition tasks, where deep neural networks are trained with a finite set of labeled training samples from K distinct known classes [5] [6] [7]. During testing, these networks trained on pre-defined K classes typically select a label from these K classes and assign the selected label to the input. Nevertheless, in most real-world classification tasks, the number of classes is usually much larger than K . Due to the limitations of various objective factors, it is commonly hard to build a dataset that covers all classes in the real-world. As a result, when a sample outside the known class set, *i.e.*, an unknown sample, is fed into the network during testing, the network will make misclassification and incorrectly recognize this sample as one of the known K classes. Many researchers have researched in unknown detection (also called anomaly detection or outlier detection) technology [8], but almost all these works are only binary classification, *i.e.*, judging whether an input is an outlier or not.

A potential solution to address this issue is Open Set Recognition (OSR) [2]. OSR assumes that deficient knowledge exists during the training stage, and aims to not only classify samples of known classes but detect unknown samples during the testing stage. As shown in Table 1.1, traditional classification tasks assume that all training and testing data come from the same label space. When a trained classifier is presented with a sample that has not been seen during the training phase, the system will wrongly recognize this sample as one of the known classes. Anomaly detection is only binary classification,

TABLE 1.1: Comparison between traditional classification, anomaly detection and open set recognition

	Traditional Classification	Anomaly Detection	Open Set Recognition
Target	Distinguish known classes	Detect outliers	Recognize known classes and detect unknown samples
Training data	Abundant data of all classes	Abundant normal data; none or few outliers	Abundant data of known classes; none unknown data
Predictions	Label of a known class	Outliers: yes or no	Label of a known class or unknown
Number of classes available during training	K	1 or 2	K
Number of classes available during testing	K	2	K+1

where the detector is trained by abundant normal data to judge whether an input is normal or anomalous. A more general technique, OSR, assumes testing samples can come from any classes, even unseen during the training phase. An open set classifier cannot only classify known classes but also detect unknown samples. The open set recognition problem can be formulated as below [2]:

$$\underset{f \in \mathcal{H}}{\operatorname{argmin}} \{ \mathcal{R}_0 + \lambda_r R_\epsilon(f(\hat{V} \cup \hat{K})) \} \quad (1.1)$$

where $\mathcal{R}_0$ is the open space risk, λ_r is a regularization constant, R_ϵ is the empirical risk function, $\hat{V}$ is the set of positive samples, and $\hat{K}$ is the set of negative samples (for details please see [2]). In this thesis, we focus on addressing OSR with deep neural networks.

1.2 Challenges and Motivations

OSR is worth studying in both the industrial field and computer vision field. For industrial tasks, OSR methods should achieve promising accuracy with a short running time. For computer vision tasks, a promising OSR method needs to achieve the highest possible accuracy on standard image datasets. In this thesis, we focus on the following inherent challenges in OSR:

- In OSR, information can only be extracted from known samples. To realize unknown detection, some OSR methods are based on generative models [9] [10] [4] [11] [12] (such as Generative Adversarial Networks (GAN) [13] or Variational

Auto-Encoders (VAE) [14]). On the other hand, some OSR methods synthesize unknown data [9] [10] [11] or redistribute the prediction scores among known classes [15] [16] [4] to directly get prediction scores of unknown samples. Although these methods achieve state-of-the-art performance in many computer vision tasks, most of them have a long running time and are not applicable to real-time industrial tasks.

- In OSR tasks, only data from known classes are provided at training time, which prevents us from training an unknown sample detector in a fully supervised manner. To realize unknown detection, many previous works in the literature model the data distribution of known samples by unsupervised learning methods [17] [18] [19] [20] [21], and those deviated data points from learned distribution are then deemed to be unknown samples. Variational Auto-encoders (VAE) [14] and Adversarial Auto-encoders (AAE) [22] are two popular probabilistic generative models used for unknown detection. Figure 1.1a and Figure 1.1b provide the visualization of the latent representations of VAE and AAE respectively in low-dimensional space when the prior distribution $p_{\theta}(z)$ is a multivariate normal Gaussian. As we can see, although these probabilistic models excel at modeling data distributions of known samples for unknown detection, it cannot provide discriminative representations to undertake classification tasks as all data follow a universal distribution.
- Class-specific features can help open set classifiers categorize known classes as well as identify unknown samples during testing [12]. AE models are well known to extract class-specific features through a pixel-level reconstruction strategy, where each pixel in the reconstructed image is expected to be exactly the same as the input image. It is usually resource-consuming due to the decoder and complex network architecture. Besides, as shown in Figure 1.2, the class activation maps indicate that an AE model weakens the feature saliency of target objects and only locates the target object well when the proportion of background is small. These limitations have rendered that such a pixel-level reconstruction strategy is less suitable for OSR tasks. As class-specific features are generally contained in target objects, not in the background, there is no need to extract features by pixel-level. It is experimentally observed in Figure 1.2 that a non-pixel-level model is more accurate in class-specific feature extraction and robust in target localization than a pixel-level reconstruction model.

FIGURE 1.1: Comparison of latent representations on the MNIST dataset of VAE (a), AAE (b), and proposed methods: CPGM-VAE (c) and CPGM-AAE (d). VAE and AAE are widely used in unknown detection, but they cannot provide discriminative features to undertake classification tasks as all features just follow one distribution. Comparatively, the proposed method can learn conditional Gaussian distributions by forcing different latent features to approximate different Gaussian models, which enables the proposed method to classify known samples as well as reject unknown samples.

1.3 Proposed OSR methods

In this thesis, we present three OSR methods based on deep neural networks. The first method focuses on shortening running time to apply OSR technology into real-world applications. The second method addresses the issue that traditional AE models cannot be used for known classification, and the third method extracts more accurate class-specific features by mutual information maximization instead of pixel-level reconstruction strategy.

In the first method, to have a shorter running time for real-world applications, we propose an OSR method, called Discriminative Loss. We combine the proposed loss function with the Softmax loss function, which is used in most Convolutional Neural Networks (CNNs), to force learned features in different classes to be close to different centroids

FIGURE 1.2: Class activation maps produced by auto-encoders and the proposed method M2IOSR. Due to the reconstruction of all pixels including target objects and background stuff, auto-encoders make the features of target objects less pronounced (*e.g.*, the first two images). By contrast, the proposed method M2IOSR can extract more class-specific features, and thus its extracted features focus more on target objects. In the images where the ratio of the background is small, *e.g.*, the third image, our method localizes the target object more accurately than auto-encoders.

for Gaussian modeling (details are introduced in Chapter 3). Those samples located in lower probability regions will be recognized as unknown.

In the second method, we propose a novel framework, Conditional Probabilistic Generative Models (CPGM), for OSR (details are introduced in Chapter 4). Different from previous anomaly detection methods based on probabilistic generative models, the proposed framework is able to generate class conditional posterior distributions $q_\phi(\mathbf{z}|\mathbf{x}, k)$ in the latent space where k is the index of known classes. These conditional distributions are forced to approximate different multivariate Gaussian models $p_\theta^{(k)}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_k, \mathbf{I})$ where $\boldsymbol{\mu}_k$ is the mean of the k -th multivariate Gaussian distribution, obtained by a fully-connected layer that maps the categorical distribution of input to the latent space. Figure 1.1c and Figure 1.1d are examples of latent representations of the proposed method

on the MNIST dataset. These learned features will be fed to an open set classifier, which consists of two parts: an unknown detector and a closed set classifier. As known samples tend to follow prior distributions, the unknown detector will recognize those samples locating in lower probability regions as unknown. Meanwhile, for the known sample, the closed set classifier will calculate its prediction scores over all known classes and predict it as the class with the highest score.

In the third method, we discard the pixel-level reconstruction strategy in AE models and pay more attention to improving the effectiveness of class-specific feature extraction. A novel OSR method, Maximal Mutual Information Open Set Recognition (M2IOSR), is thus proposed (details are introduced in Chapter 5). Different from most previous deep learning-based OSR methods [9] [10] [4] [11] [12], M2IOSR has a streamlined architecture and only uses an encoder to discover class-specific features by maximizing the mutual information (MI) [23] between the input and its latent features. The mutual information greatly helps the encoder establish stronger interdependence between the input image and its learned features for OSR. Meanwhile, to further reduce the open space risk [2], this model also encourages latent features to have desired statistical characteristics. For statistical constraint, we replace the adversarial learning in [24] by the KL-divergence loss function proposed in the second proposed method CPGM.

1.4 Contributions

In a summary, in this thesis, our main contributions can be stated as follows:

- We develop an online air leakage detection system of a pneumatic train door subsystem using the proposed Discriminative Loss method. In the proposed method, the loss function within the CNN is modified to force learned features in different classes to be close to different centroids for Gaussian modeling. The proposed method is demonstrated on the MNIST dataset and applied to detect different levels of air leakage on the pneumatic train door subsystem and achieves promising accuracy with a shorter running time.
- We propose a novel framework, called Conditional Probabilistic Generative Models (CPGM), for OSR. Compared with the original VAE and AAE models, CPGM is able to learn class conditional Gaussian distributions for known classification

and unknown detection. We also explore many model variants of CPGM and present much practical guidance to use VAE and AAE for OSR. Experiments on several benchmark datasets show that our method brings 9.96% OSR performance gain on the MNIST-based dataset and brings 7.99% OSR performance gain on the LSUN dataset.

- We propose a novel OSR method, Maximal Mutual Information Open Set Recognition (M2IOSR). Different from most previous deep learning-based OSR methods, the proposed method only uses an encoder architecture to extract class-specific features for OSR. In this method, to further reduce open space risk, we adopt a KL-divergence loss function to encourage latent features to follow class conditional Gaussian distributions. Meanwhile, to prevent MI from being lost in the middle layers, we incorporate multi-scaled local MI maximization. Experiments on several benchmarks demonstrate that the proposed method significantly enhances the performance of our baselines and achieves new state-of-the-art results on several benchmarks consistently.
- We compare the proposed three methods on six standard image datasets and the air pressure data collected from a pneumatic train door subsystem. All the proposed methods do not significantly decline closed set classification accuracies while realizing OSR. For experiments on standard image datasets, CPGM significantly outperforms the baseline methods, and M2IOSR achieves state-of-the-art performance. For experiments on air pressure data, Gaussian-based models (Discriminative Loss, CPGM-VAE, and CPGM-AAE) all achieve promising results. Among them, CPGM-AAE achieves the highest F_1 scores in both the extension phase and the retraction phase. It is also worth noting that compared with other OSR methods, Discriminative Loss has a minimal number of parameters, and its running time is only 0.2 milliseconds per input slower than that of the fastest method (Softmax).

1.5 Outline of the Thesis

The rest of this thesis is organized as follows:

Chapter 2 reviews deep learning models used in proposed methods, existing OSR methods, deep learning-based unknown detection, and mutual information for unsupervised learning.

Chapter 3 introduces the proposed Discriminative Loss method, which has a shorter running time. This method is demonstrated on the MNIST dataset and the air pressure data collected from a pneumatic train door subsystem.

Chapter 4 presents a novel OSR framework, Conditional Probabilistic Generative Models (CPGM), and shows its performance on six standard image datasets.

Chapter 5 demonstrates a novel OSR method, Maximal Mutual Information Open Set Recognition (M2IOSR), as well as its results on several benchmarks.

Chapter 6 compares the proposed three methods on six standard image datasets and the air pressure data collected from a pneumatic train door subsystem.

Chapter 7 concludes this thesis and discusses future research works.

Chapter 2

Literature Review

In this chapter, we review and discuss previous works that are related to this thesis from different perspectives. Firstly, we briefly review deep learning models used in proposed methods in this thesis, including Visual Geometry Group (VGG) neural networks, Variational Auto-encoders (VAE), and Adversarial Auto-encoders (AAE). Secondly, we review and compare existing open set recognition (OSR) methods. Next, we briefly review deep learning-based unknown detection methods and mutual information (MI) theory used in unsupervised learning.

2.1 Deep Learning Models

In this section, we briefly review some classical deep learning models (*i.e.*, VGG neural networks, Variational Auto-encoders, and Adversarial Auto-encoders) that used in the proposed methods in this thesis.

2.1.1 VGG Neural Networks

Karen Simonyan and Andrew Zisserman proposed the VGG neural network [6], which is a Convolutional Neural Network (CNN) architecture. Different from previous classical CNNs (*e.g.* AlexNet [25]) that focused on smaller strides and window sizes, VGG addresses another very important aspect of CNNs: depth. An example of VGG neural networks, VGG-13, which has 13 convolutional layers and 3 fully-connected layers, is

FIGURE 2.1: Architecture of VGG-13.

shown in Figure 2.1. The input of VGG is fixed to a 224×224 pixel Red Green Blue (RGB) image, thus sometimes crop or resize operations need to be conducted to make the input image size consistent. VGG uses several consecutive 3×3 convolution kernels, which is the smallest size that still can capture up/down and left/right information, to replace the larger convolution kernels used in previous CNNs. Using such stacked small convolution kernels is better than just using a large convolution kernel because multiple non-linear layers can increase the depth of the network to ensure that more complex patterns can be learned, and the cost is relatively small. VGG also used 1×1 convolution filters to realize linear transformation, which is followed by a Rectified Linear Unit (ReLU). The convolution stride is fixed to 1 so that the spatial resolution can be preserved after convolution.

Generally, with the small-size convolution filters, VGG has a large number of weight layers and improves its power to extract robust features. Hence, in the proposed methods, we employ VGG neural networks as the backbone network for OSR.

2.1.2 Variational Auto-encoders

The Variational Auto-encoders (VAE) [14] commonly consists of an encoder, a decoder and a loss function $\mathcal{L}(\theta; \phi; x)$. The encoder is a neural network that has parameters

FIGURE 2.2: Framework of VAE when the prior distribution is the normal distribution.

ϕ . Its input is a sample $\mathbf{x}$ and its output is a hidden representation $\mathbf{z}$. The decoder is another neural network with parameters θ . Its input is the representation $\mathbf{z}$ and it outputs the probability distribution of the sample. The loss function in the VAE is defined as follows:

$$\begin{aligned} \mathcal{L}(\theta; \phi; \mathbf{x}) = & -D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}) || p_{\theta}(\mathbf{z})) \\ & + \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})] \end{aligned} \quad (2.1)$$

where $q_{\phi}(\mathbf{z}|\mathbf{x})$ is the approximate posterior, $p_{\theta}(\mathbf{z})$ is the prior distribution of the latent representation $\mathbf{z}$ and $p_{\theta}(\mathbf{x}|\mathbf{z})$ is the likelihood of the input $\mathbf{x}$ given latent representation $\mathbf{z}$. On the right-hand side of Eqn. (2.1), the first term is the KL-divergence between the approximate posterior and the prior. It can be viewed as a regularizer to force the approximate posterior to be close to the prior $p_{\theta}(\mathbf{z})$. The second term can be viewed as the reconstruction errors to force the output to reconstruct the input correctly. The framework of VAE when the prior distribution is the normal distribution is shown in Figure 2.2: this VAE model cannot only reconstruct the input image correctly, but also extract latent features with normal distribution.

Commonly, the prior over the latent representation $\mathbf{z}$ is the centered isotropic multivariate Gaussian $p_{\theta}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$. In this case, the variational approximate posterior could be a multivariate Gaussian with a diagonal covariance structure:

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \mathbf{I}) \quad (2.2)$$

where the mean $\boldsymbol{\mu}$ and the standard deviation $\boldsymbol{\sigma}$ of the approximate posterior are outputs of the encoding multi-layered perceptrons (MLPs). The latent representation $\mathbf{z}$ is defined as $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ is the element-wise product. Let J be the dimensionality of $\mathbf{z}$, then the KL-divergence can be calculated [14]:

$$\begin{aligned} -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z})) \\ = \frac{1}{2} \sum_{j=1}^J (1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2) \end{aligned} \quad (2.3)$$

With the loss function $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{\phi}; \mathbf{x})$, the VAE is trained not only to reconstruct the input accurately, but also to force the posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ in the latent space to approximate the prior distribution $p_\theta(\mathbf{z})$. If a sample locates in the low probability region of the learned distribution, this sample will be recognized as unknown.

2.1.3 Adversarial Auto-encoders

The Adversarial Auto-encoders (AAE) [22] model is composed of three modules: an encoder (generator) $G(\mathbf{x})$, a decoder $De(\mathbf{z})$ and a discriminator $D(p(\mathbf{z}))$. The AAE model is trained in two phases: the *reconstruction* phase and the *regularization* phase. In the *reconstruction* phase, the encoder $G(\mathbf{x})$ and decoder $De(\mathbf{z})$ are trained jointly to reconstruct the input $\mathbf{x}$ correctly, which can be expressed as follows:

$$\mathcal{L}_r = \|\mathbf{x} - \tilde{\mathbf{x}}\|_2 \quad (2.4)$$

where $\tilde{\mathbf{x}}$ is the reconstructed image.

The framework of AAE when the prior distribution is the normal distribution is shown in Figure 2.3. In the *regularization* phase, an adversarial game is established between the encoder $G(\mathbf{x})$ and the discriminator $D(p(\mathbf{z}))$: firstly the discriminator $D(p(\mathbf{z}))$ is trained to distinguish the true distribution (generated using the prior distribution $p(\mathbf{z})$) from the posterior distribution $q(\mathbf{z}|\mathbf{x})$ (generated by the encoder $G(\mathbf{x})$), then the encoder $G(\mathbf{x})$ is trained to confuse the discriminator $D(p(\mathbf{z}))$. This adversarial process can be

FIGURE 2.3: Framework of AAE when the prior distribution is the normal distribution.

expressed in the following equation:

$$\min_G \max_D E_{p(z)} [\log D(p(z))] + E_{x \sim p_{data}} [\log(1 - D(G(x)))] \quad (2.5)$$

When training is completed, the posterior distribution $q(z|x)$ in the latent space will be close to the prior distribution $p(z)$. If a sample locates in the low probability region of the learned distribution, this sample will be recognized as unknown.

2.2 Open Set Recognition

In [2], the concept of open set recognition (OSR) was put forward and an SVM-based OSR method (1-vs-Set machine) was introduced, which added an extra hyper-line to separate unknown samples from known samples. WSVM [26] and P_I -SVM [27] improved this 1-vs-Set machine by introducing the extreme value theory into the SVM to further control the open space risk. Cevikalp *et al.* [28] [29] defined the acceptance regions for known samples with a family of quasi-linear “polyhedral conic” functions. SROSR [30] emphasized on extracting useful information from reconstruction error distributions. Some distance-based OSR methods were also proposed. NNO [31] recognized those samples away from the centroids of the known classes as unknown. NNDR [32] classified known classes and detected unknown samples by evaluating similarity scores between the two most similar classes.

The above methods are mostly based on traditional machine learning techniques (*e.g.*, support vector machine, sparse representation, nearest neighbor, etc.). With deep learning achieving outstanding results in many computer vision fields, in recent years, more and more researchers focus on using deep learning techniques to address OSR tasks. One straightforward OSR method is setting a threshold on the probability of the most probable class produced by the Softmax layer [33], but unknown samples could also produce equally high probabilities [2]. Openmax [16] overcame this shortcoming by redistributing the probability distribution produced from the Softmax function to get the predicted probability of the unknown class. Similarly, Shu *et al.* [15] proposed the Deep Open Classifier (DOC) model, which replaces the Softmax layer with a 1-vs-rest layer containing sigmoid functions. Differently, some works directly estimated probabilities of unknown samples by adding synthesized unknown-class images into the training set, such as G-Openmax [9] and OSRCI [10]. Besides, a hybrid model proposed in [34] combined an AE model, a classifier, and a density estimator to detect unknowns. RPL [35] classified known and unknown samples by the embedding of other samples with reciprocal points. Auto-encoders (AE) are also widely used in OSR tasks. CROSR [4] combined predicted scores and latent features to get probabilities of unknown samples. C2AE [11] determined the decision boundary by using the extreme value theory on reconstruction errors. GDFR [12] extracted optimal features by feeding the input and its reconstruction into a self-supervised learning model.

2.3 Unknown Detection

Unknown detection (also called outlier detection or anomaly detection) aims to distinguish anomalous samples from normal samples. Some general anomaly detection methods are based on Support Vector Machine (SVM) [36] [37] or forests [38]. In recent years, deep neural networks have also been used in anomaly detection, mainly based on auto-encoders trained in an unsupervised manner [19] [17] [18] [20]. Auto-encoders commonly have a bottleneck architecture to induce the network to learn abstract latent representations. Meanwhile, these networks are typically trained by minimizing reconstruction errors. In anomaly detection, the training samples commonly come from the same distribution, thus the well-trained auto-encoders could extract the common latent

representations from the normal samples and reconstruct them correctly, while anomalous samples do not contain these common latent representations and could not be reconstructed correctly. Although VAE is widely applied in anomaly detection, it cannot provide discriminative features for classification tasks.

Apart from auto-encoders, some studies used Generative Adversarial Networks (GANs) to detect anomalies [39] [21]. GANs are trained to generate similar samples according to the training samples. Given a testing sample, the GAN tries to find the point in the generator's latent space that can generate a sample closest to the input. Intuitively, the well-trained GAN could give good representations for normal samples and terrible representations for anomalies.

There are also some related topics that focus on handling novel classes. For example, incremental learning [40] aims to make predictions on both old classes and new classes without accessing data in old classes. Few-shot learning [41] [42] [43] [44] aims to utilize a network trained on old classes to undertake vision tasks on unseen classes.

2.4 Mutual Information for Unsupervised Learning

In information theory, the Mutual Information (MI) $I(A; B)$ is a measure of the mutual dependence between the two variables A and B . More specifically, it quantifies the “amount of information” about the variable A (or B) captured via observing the other variable B (or A). The MI can be expressed as the difference between two entropy terms:

$$I(A; B) = H(A) - H(A|B) = H(B) - H(B|A), \quad (2.6)$$

where $H(A)$ and $H(B)$ are the marginal entropy of A and B . This definition has an intuitive interpretation: $I(A; B)$ is the reduction of uncertainty in one variable when the other variable is observed.

If A and B are independent to each other, their MI is zero, *i.e.* $I(A; B) = 0$. In this case, knowing A does not give any information about B and vice versa. At the other extreme, if A and B are interdependent by a deterministic function, then all information conveyed by A is shared with B . Thus, knowing A determines the value of B and vice versa, where the maximum MI is attained. In the case of jointly continuous random variables, the MI can be expressed as the following formulation:

$$\begin{aligned}
I(A; B) &= \int \int p(a, b) \log \frac{p(a, b)}{p(a)p(b)} da db \\
&= \int \int p(b|a)p(a) \log \frac{p(b|a)}{p(b)} da db,
\end{aligned} \tag{2.7}$$

where $p(a, b)$ is the joint probability density function of A and B , and $p(a)$ and $p(b)$ are the marginal probability density functions of A and B respectively.

Mutual information (MI) applied in neural networks for unsupervised learning has a long history [45] [46] [47] [48]. Recently, MI is widely used in deep neural networks for unsupervised discrete representation learning [49], scene representation learning [50], segmentation and clustering [51]. Especially, MINE [52] learned an MI estimate of continuous variables based on generative models. DIM [24] followed MINE in this respect, but emphasized the importance of maximizing local MI, which was also mentioned in other literature [53] [54]. Some researchers also study on finding better bounds on MI used in deep learning [55] [56].

Chapter 3

Open Set Recognition with Short Running Time

OSR methods are gaining more and more attention these years [2] [16] [26] [27] [30] [31] [32] [9] [15] [10] [4] [11] [12] [35] [34]. Considering that information can only be extracted from known samples, to realize unknown detection, some OSR methods are based on generative models [9] [10] [4] [11] [12] (such as Generative Adversarial Networks (GAN) [13] or Variational Auto-Encoders (VAE) [14]). On the other hand, some OSR methods synthesize unknown data [9] [10] [11] or redistribute the prediction scores among known classes [15] [16] [4] to directly get prediction scores of unknown samples. Although these methods achieve state-of-the-art performance in many computer vision tasks, most of them have a long running time and are not applicable to real-time industrial tasks. To have a shorter running time, we proposed an OSR method, called Discriminative Loss. We combine the proposed loss function with the Softmax loss function, which is used in most Convolutional Neural Networks (CNNs), to force learned features in different classes to be close to different centroids for Gaussian modeling. The proposed method is demonstrated on the MNIST [57] dataset. This method is also applied to detect air leakage on pneumatic train door subsystems and achieves promising classification accuracy with a shorter running time. In Chapter 6, Discriminative Loss is also compared with other proposed methods on six standard image distastes.

3.1 Introduction

In the industrial field, a traditional method for fault detection is based on a manual inspection at regular intervals, which is time consuming, laborious, and also relies on the experience of the trained personnel. With the development and deployment of low-cost sensors, signal-based methods are widely used for fault detection. Signal-based methods realize fault detection by observing the changes of the monitored one or more signals based on various sensors, but the analysis of the monitored signal curve and the reference one is conducted manually by skilled experts and the final decision is made in a subjective manner. To address the shortcomings of signal-based methods, in recent years, artificial intelligence (AI)-based methods for fault detection are being received more and more attention. The AI-based methods can be further divided into two categories: machine learning methods and deep learning methods. The machine learning methods are required to manually extract and select features for fault detection. These features may be correlated or redundant, resulting in long training time and model over-fitting. Compared to machine learning methods, deep learning methods are able to automatically extract high-quality features from a large amount of sensor data [58] and need less human labor or expert knowledge for feature selection.

Using deep learning for fault detection can be viewed as a type of classification task. A large collection of known fault samples is necessary to generate a reliable deep learning model. The learned model will perform very well when presented with data that belong to the trained categories but may misclassify when data that does not belong to the trained categories are presented. In real-world applications, due to the limitations of various objective factors, it is very difficult to pre-collect all possible types of fault or anomaly conditions to train the model. Many researchers have researched in unknown detection (also called anomaly detection or outlier detection) technology [8] and successfully applied it in the industrial field [59] [60] [61] [62] [63], but almost all these works are only binary classification, i.e., judging whether an input is an outlier. A more general technique, called Open Set Recognition (OSR)[2], assumes testing samples can come from any classes, even unseen during the training phase. An open set classifier not only can classify known classes but also can detect unknown samples.

OSR methods are gaining more and more attention these years [16] [9] [10] [4] [11] [15]. Considering that information can only be extracted from known samples, to realize unknown detection, most OSR methods are based on generative models [9] [10] [4]

[11] [15] (such as Generative Adversarial Networks (GAN) [13] or Variational Auto-Encoders (VAE) [14]). These generative models are usually complex, needing advanced computational models and a long time for training and testing [64]. Meanwhile, On the other hand, to get the prediction score of the unknown class, some methods need to synthesize unknown data [9] [10] [11] or redistribute the prediction scores among known classes [16] [4] [15]. Therefore, although these methods achieve state-of-the-art performance in many computer vision tasks, they are difficult to apply to real-time industrial tasks. In this paper, we propose an OSR method based on a traditional CNN with a shorter running time. The proposed method is demonstrated on the MNIST dataset and applied to detect air leakages on the pneumatic train door subsystem. The developed air leakage detection system not only can accurately classify known air leakage faults but also can raise an alarm when meeting unknown faults.

3.2 Methodology

As is known that in the majority of deep neural networks for classification, the output of the penultimate layer is fed to the Softmax function which produces the probabilities of input belonging to each known class, and then the class with the maximum probability will be selected as the output. The Softmax function

$$S_k = \frac{e^{a_k}}{\sum_{j=1}^n e^{a_j}} = \frac{e^{\phi(W_k^T \mathbf{f} + b_k)}}{\sum_{j=1}^n e^{\phi(W_j^T \mathbf{f} + b_j)}} \quad (3.1)$$

gives the probability of the input sample belonging to the k -th class. Here $\mathbf{f} \in \mathbb{R}^d$ is the output of the penultimate layer, $\phi(\cdot)$ is an activation function, $W_j^T \in \mathbb{R}^d$ denotes the j -th row of the weights $W^T \in \mathbb{R}^{n \times d}$ in the last layer, and b_k is the k -th element of the bias term $b \in \mathbb{R}^n$. Considering that networks always choose the class with the maximum probability as the final decision, the training strategy is to make the target class's probability S_d maximum. This is achieved by minimizing the softmax loss function:

$$L_S = - \sum_{i=1}^m \ln S_d^{(i)} \quad (3.2)$$

where m is the size of a mini-batch.

FIGURE 3.1: Comparison of latent features learned from a LeNet model (a CNN model) on the MNIST dataset between using the Softmax loss (a) and using the proposed Discriminative Loss (b).

To have a more intuitive understanding of the Softmax loss function, we experiment on the MNIST dataset [57] by training a LeNet model (a CNN model) [65]. The number of neurons in the last fully-connected layer is reduced to 2. The output of the fully-connected layer can be viewed as the extracted features of the deep neural network. When the training loss converges, all two-dimensional features from the training set are plotted on a plane as shown in Figure 3.1a. The X axis is the output of the first neuron and the Y axis is the output of the second neuron. From Figure 3.1a, the extracted features from different classes are separable and fall within the corresponding region. The magnitude of each feature does not affect the assignment of its class. This is the reason why these distributions present a ‘radial’, which has been proved in [66].

As is shown in Figure 3.1a that the Softmax function divides the whole feature space into several sub-spaces to make different features separable. This training strategy can be viewed as a kind of linear classification, which is effective for known classification (the testing accuracy is around 97.5% on the MNIST dataset), but not enough for OSR, as the decision boundaries of these distributions are difficult to determine for unknown detection. To realize OSR, the decision boundary of each class needs to be described, which is called domain description [67]. We hope the learned features not only to be separable but also to be discriminative, i.e., each feature distribution should have small intra-class distances and large inter-class distances for Gaussian modeling (see Figure 3.2). Gaussian modeling is a popular method in unknown detection [8] as the Gaussian distribution could give a relatively clear decision boundary to judge whether a sample is normal or anomalous.

FIGURE 3.2: The difference between linear classification, domain description, and discriminative features.

As hinted in [66], the loss function affects the feature distribution. Hence, one method to obtain discriminative features is to modify the loss function to change the feature distribution. Here, a new loss function is proposed to minimize intra-class distances, which is given as follows:

$$L_D = \frac{1}{2m} \sum_{i=1}^m \mathbf{d}_{y_i}^T \mathbf{d}_{y_i} \quad (3.3)$$

where m is the size of a mini-batch. $\mathbf{d}_{y_i}^T \mathbf{d}_{y_i}$ describes the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{c}_{y_i}$, where $\mathbf{d}_{y_i} = \mathbf{x}_i - \mathbf{c}_{y_i}$ is the difference between $\mathbf{x}_i$ and $\mathbf{c}_{y_i}$. $\mathbf{x}_i \in \mathbb{R}^d$ is the learned feature of i -th sample belonging to the y_i -th class, and $\mathbf{c}_{y_i} \in \mathbb{R}^d$ is the centroid of y_i -th class. Meanwhile, we adopt the joint supervision of the proposed loss function and Softmax loss function as below:

$$L = L_S + \lambda L_D \quad (3.4)$$

where λ is a scalar used for balancing these two loss functions. The training details of the proposed method are summarised in Algorithm 1. To have a more intuitive feeling about the proposed method, we also experiment on the MNIST dataset [57] by training a LeNet model [65]. The only difference with the previous experiment is that the proposed joint loss function (Equation (3.4)) is adopted to train the network. In this experiment, λ is set to 0.5. When the training loss converges, all two-dimensional features from the training set are plotted on a plane as shown in Figure 3.1b. Compared with the result in the previous experiment (Figure 3.1a), the learned features in the same class are much more compact and away from other classes, therefore each feature distribution can be expressed by a multivariate Gaussian model $\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2)$ for unknown detection, where $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k^2$ are the mean and variance of k -th class features respectively. We also investigate the sensitivity of the balancing parameter λ . The balancing parameter

FIGURE 3.3: Testing accuracies using Discriminative Loss on the MNIST dataset with different λ .

λ is varied from 0 to 1 and the testing accuracies on the MNIST dataset are shown in Figure 3.3. The proposed method can improve the performance on the MNIST dataset over a wide range of λ .

Algorithm 1: Training procedure.

Input: (X, Y) - Training data X and labels Y .

Require: Initialized parameters θ_0 in the network G .

Initialize: Set the number of iteration $t = 0$.

- 1: **while** training loss dose not converge **do**:
- 2: Sample a mini-batch (X_{batch}, Y_{batch}) from (X, Y) .
- 3: Update the number of iteration: $t = t + 1$.
- 4: Compute the learned features: $F_{batch} = g(X_{batch})$.
- 5: Compute the joint loss: $L^t = L_S^t + \lambda L_D^t$.
- 6: Update the centroids c_k for each class k : $c_k^{t+1} = \frac{1}{2} [c_k^t + mean(F_{batch.k})]$.
- 7: Update the parameters θ of G using mini-batch gradient descent (MBGD):
 $\theta^{t+1} = \theta^t - \mu^t \frac{\partial L^t}{\partial \theta^t}$.
- 8: **end while**

Output: Centroids c_k and variance σ_k^2 for each class.

Output: Learned model G , with learned parameters θ .

When training is completed, the per class multivariate Gaussian distribution $f_k(z) = \mathcal{N}(z; \mu_k, \sigma_k^2)$ from all training samples is modeled. During testing, if the dimension of the learned feature z is n : $z = (z_1, \dots, z_n)$, the probability of this sample locating in the distribution $f_k(z)$ is defined as follows:

$$P_k(z) = 1 - \int_{\mu_1 - |z_1 - \mu_1|}^{\mu_1 + |z_1 - \mu_1|} \dots \int_{\mu_n - |z_n - \mu_n|}^{\mu_n + |z_n - \mu_n|} f_k(t) dt \quad (3.5)$$

Those testing samples locating in the low probability regions will be recognized as unknown. The decision threshold is defined as τ , if $\forall k \in \{1, \dots, K\}$, $P_k(\mathbf{z}) < \tau$ satisfies (where N is the number of known classes), then this sample will be classified as unknown, otherwise, this sample will be recognized as known with the predicted class from a Softmax layer. The above testing procedure is summarized in Algorithm 2. In the next section, the experiments of the proposed method on collected air pressure data are described.

Algorithm 2: Testing procedure.

Require: Testing sample $\mathbf{X}$.

Require: Centroids $\mathbf{c}_k$ and variance σ_k^2 for each class.

Require: Learned model $\mathbf{G}$.

Require: Threshold τ of Gaussian distributions.

- 1: Compute the latent feature of input: $\mathbf{z} = \mathbf{G}(\mathbf{X})$.
- 2: Compute predicted known label y_{pred} of input $\mathbf{X}$: $y_{pred} = \text{argmax}(\text{Softmax}(\mathbf{z}))$.
- 3: **if** $\forall k \in \{1, \dots, K\}$, $P_k(\mathbf{z}) < \tau$ **then:**
- 4: predict $\mathbf{X}$ as unknown;
- 5: **else:**
- 6: predict $\mathbf{X}$ as known with label y_{pred} .
- 7: **end if**

Output: Predicted label of $\mathbf{X}$: y_{pred} or *unknown*.

3.3 The Pneumatic Train Door Subsystem

The train door subsystem serves as a key component in the railway system. It has been reported that around 30% to 40% of train service delays or failures are related to train door faults [68] [69]. Once train door faults happen, the train has to be taken off service, causing economic loss and travel delays, sometimes even accidents. For pneumatically operated train doors, air leakage will cause gradual degradation of train door performance and may lead to train service disruption [70]. Hence, continuous monitoring, and earlier detection of air leakages to alert train operators of imminent door faults would greatly reduce train downtime and ensure both safety and high availability of the railway system. Although existing OSR methods achieve state-of-the-art performance in computer vision tasks, they are difficult to apply to this real-time industrial task: 1) score redistribution method divides the classification process into two stages: score generation and score redistribution, which does not meet the requirements of real-time monitoring

FIGURE 3.4: A test jig replicating the actual pneumatic train door subsystem. The actuator is marked with a green border, the pressure transducer is marked with a blue border, and the embedded controller is marked with a red border.

in industrial applications; 2) time-series sensor data do not contain as various information as images (commonly used in computer vision tasks) for unknown data synthesis; 3) GAN and VAE these generative models are usually complex, needing advanced computational devices and a long time for training and testing. Considering the above discussions, to apply OSR to industrial applications, it is desired the method is an end-to-end system and can realize OSR by trained only on known data with a streamlined and lightweight network (such as Convolutional Neural Network (CNN)).

To collect enough air pressure data, an exact replica of the actual pneumatic train door subsystem has been built in the SMRT-NTU Smart Urban Rail Corporate Laboratory, Singapore (Figure 3.4). In this section, we describe the test jig, as well as the data acquisition and preparation process. The flowchart is depicted in Figure 3.5.

3.3.1 Pneumatic Module

The door movement control (opening and closing doors) is achieved by a pneumatic module and its schematic illustration is shown in Figure 3.6. The basic components in the pneumatic module are the pneumatic actuator, an electromagnetic valve (EMV), and a manifold chamber connected to the air compressor. The pneumatic actuator mainly consists of a cylinder with a piston and the compressed air inside the chamber will force the piston to move in the desired direction. The EMV is used to control the flow direction of the compressed air. Whenever a 110V voltage is applied, the EMV will be energized and channeled the compressed air into the right chamber. The compressed air in the

FIGURE 3.5: The framework of the proposed online air leakage detection system. This system could detect different levels of air leakage through the following three steps: data acquisition, signal processing, and air leakage detection based on the proposed OSR method.

right chamber will force the piston to extend to the left side, causing the door to open (the direction of airflow is shown by the red line in Figure 3.6). On the other hand, when the applied 110V voltage is removed, the direction of the flow of the compressed air will be reversed and air will be channeled into the left chamber. Hence the piston will retract to the right and the door will be closed (the direction of airflow is shown by the green line in Figure 3.6).

A pressure transducer is installed in the inlet to sense the air pressure. This pressure transducer is wired to an embedded controller. From the controller, the air pressure data can be wirelessly collected on a laptop or mobile phone. The pressure data are collected at the interval of 0.1 seconds, during the two distinct phases of door operations: the extension phase (door opening) and the retraction phase (door closing). A typical pressure signal collected is shown in Figure 3.7.

FIGURE 3.6: The schematic illustration of the pneumatic module used to control the door movement. The direction of airflow when door opening is shown by the red line, while the direction of airflow when door closing is shown by the green line.

FIGURE 3.7: A typical pressure signal in one cycle, which is made up of two phases: extension (door opening) and retraction (door closing). The extension signal and retraction signal are similar but not exactly the same.

3.3.2 Data Acquisition

Over time, minute air leakages would develop but might not show up as door faults. The main purpose of this work is to develop a condition monitoring system to detect these minute air leakages that may lead to imminent train door failure.

A large amount of data are required for the training of a deep learning model. However, it is impossible to collect enough data to cover all operational scenarios. Based on operational experiences, four levels of air leakages: (*leakage 1, 2, 3, and 4*) were defined, in addition to normal operational data, i.e., no leakage.

Figure 3.7 shows a typical air pressure signal collected by a pressure transducer. The data have been filtered and normalized to detect air leakages. We treat the door opening data, and door closing data separately, so that we have two similar classifiers, one for door opening and another for door closing. The plots of the five classes of data (filtered

FIGURE 3.8: Air pressure signals collected in five conditions (*normal*, *leakage 1*, *2*, *3*, and *4*) after mean filtering and normalization in the extension phase (a) and retraction phase (b).

and normalized) are shown in Figure 3.8. It can be seen that it could be quite challenging to distinguish between the various levels of air leakage by traditional signal processing techniques, especially for the extension phase (see Figure 3.8a).

The leakage scenarios represent minute leaks that will not affect the normal operation of the train doors but prolonging such leakages may cause eventual train door faults, and hence hard to be detected by conventional condition monitoring or fault detection methods. Thus it is desirable to develop a condition monitoring system to pick up these leakages and alert the train operator pre-emptively before such leakages develop into actual door faults causing train delays or even withdrawals.

3.3.3 Data Processing

To test our proposed OSR method, we used *leakage 1*, *2*, *3* and the normal operating condition (i.e., no leakage) to represent scenarios where data can be collected while *leakage 4* as a scenario where data cannot be collected. Hence, *leakage 4* data are not used in the training of the model but used in testing the proposed OSR method. A CNN is selected as the learning model to extract useful information from collected data. To feed collected air pressure data into the CNN, a 2D transformation is adopted. The collected air pressure data are segmented and the number of sampling points per segment is kept as consistent. Then each segment x is divided into m equal parts and each part contains n sampling points. Finally, each part is taken as each line of the signal image sequentially. This 2D transformation can be described in the following mathematical format:

FIGURE 3.9: The process of generating a signal image by the 2D transformation.

$$\mathbf{I} = \begin{bmatrix} x(t_0) & \cdots & x(t_0 + n - 1) \\ \vdots & \ddots & \vdots \\ x(t_0 + (m - 1)n) & \cdots & x(t_0 + mn - 1) \end{bmatrix} \quad (3.6)$$

where $\mathbf{I}$ is the signal image and $x(t_0)$ is the signal data of time t_0 . As mentioned before, each phase of the door cycle contains 100 sampling points. For each sample, 8 phases of data are collected and the last 16 sampling points are discarded. Then these 784 sampling points are converted into a 28x28 image (see Figure 3.9). A total of 51,200 samples are collected from *normal*, *leakage 1*, *2*, and *3* these four situations, and random 38,400 samples of them are chosen as the training set while the remaining 12,800 samples are chosen as the testing set. Meanwhile, 12,800 samples are collected under the *leakage 4* situation as unknown samples. They are then added to the testing set to constitute an open testing set.

3.4 Experiments and Results

3.4.1 Implementation Details

The experiments were conducted under the developed online air leakage detection system. The proposed method is used to detect different levels of air leakage and unknown air leakage faults. The parameter λ in the proposed loss function (Equation (3.4)) is set equal to 0.5. The CNN used here is a VGG [6] inspired version, denoted as VGG-7, as

TABLE 3.1: The parameters in the constructed CNN.

Input Size	Layer Name	Kernel Size	Padding	Output Size
28*28*1	Conv1_1	5*5*32	2	28*28*32
28*28*32	Conv1_2	5*5*32	2	28*28*32
Max-Pooling				
ReLU				
14*14*32	Conv2_1	5*5*64	2	14*14*64
14*14*64	Conv2_2	5*5*64	2	14*14*64
Max-Pooling				
ReLU				
7*7*64	Conv3_1	5*5*128	2	7*7*128
7*7*128	Conv3_2	5*5*128	2	7*7*128
Max-Pooling				
ReLU				
Fully-Connected (3*3*128 $\rightarrow$ 50)				
ReLU				
Fully-Connected (50 $\rightarrow$ 2)				
Fully-Connected (2 $\rightarrow$ 4)

FIGURE 3.10: The architecture of the constructed CNN (**C**: convolutional layer; **M**: max-pooling layer; **R**: ReLU layer; **F**: fully-connected layer).

shown in Figure 3.10. The network parameters are listed in Table 3.1, and this network can be simply described as "Input(32*32) + 2*(32C28) + MP2 + 2*(64C14) + MP2 + 2*(128C7) + MP2 + FC50 + FC2 + FC4". During training, an Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.001 is used and the batch size is fixed to 64. The decision threshold τ is set to 0.03 (according to the '68-95-99.7' rule [71]).

3.4.2 Air Leakage Classification

Figure 3.11 and Table 3.2 compare the ability of our proposed method in classifying *known* samples. Figure 3.11a (extension phase) and Figure 3.11b (retraction phase) show the classification results when only Softmax loss function is used, i.e. $\lambda = 0$ in Equation (3.4). Although the learned features are separable in the extension phase, the intra-class distances are too large to determine the decision boundaries for unknown

TABLE 3.2: Comparison of closed set testing accuracy on air pressure data between using the Softmax loss function and using the proposed loss function.

Method	Extension	Retraction
Softmax Loss	98.80%	80.15%
Discriminative Loss	98.90%	98.48%

detection, especially in the *normal* and *leakage 1* conditions. In the retraction phase, most learned features are overlapped and this is the reason why the closed set testing accuracy is 98.80% for the extension phase but only 80.15% for the retraction phase. Figure 3.11c (extension phase) and Figure 3.11d (retraction phase) show the classification results when the proposed loss function, i.e. Equation (3.4) with $\lambda = 0.5$ is used. The learned features are compact (small intra-class distance) and separable (large inter-class distance). The closed set testing accuracy is 98.90% for the extension phase and 98.48% for the retraction phase.

FIGURE 3.11: Comparison of latent features learned from the constructed CNN on the pressure training dataset between using the Softmax loss function (extension phase (a), retraction phase (b)) and using Discriminative Loss where λ is set equal to 0.5 (extension phase (c), retraction phase (d)).

3.4.3 OSR on Air Pressure Data

In this section, we examine the utility of the proposed OSR method on the air pressure dataset and compare it with other OSR methods from accuracy and efficiency. To evaluate the accuracy of OSR, here we use the F-measure values (Macro- F_1 and Micro- F_1), which are defined as follows [72]:

$$\text{precision}_a = \frac{\sum_{k=1}^N \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}}{N} \quad (3.7)$$

$$\text{recall}_a = \frac{\sum_{k=1}^N \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}}{N} \quad (3.8)$$

$$\text{precision}_i = \frac{\sum_{k=1}^N \text{TP}_i}{\sum_{k=1}^N (\text{TP}_i + \text{FP}_i)} \quad (3.9)$$

$$\text{recall}_i = \frac{\sum_{k=1}^N \text{TP}_i}{\sum_{k=1}^N (\text{TP}_i + \text{FN}_i)} \quad (3.10)$$

$$\text{Macro-}F_1 = \frac{2 \times \text{precision}_a \times \text{recall}_a}{\text{precision}_a + \text{recall}_a} \quad (3.11)$$

$$\text{Micro-}F_1 = \frac{2 \times \text{precision}_i \times \text{recall}_i}{\text{precision}_i + \text{recall}_i} \quad (3.12)$$

where k is the index of class and N is the number of available classes for training. TP, FP, and FN stand for the number of true positive, false positive, and false negative samples, respectively.

As mentioned before, the testing dataset contains 12,800 unknown samples and 12,800 known samples. All testing samples are fed into the trained CNN using the proposed loss function. The learned features are shown in Figure 3.12. As can be seen from Figure 3.12, in both phases, known-class features are clustered into four separate clusters and each cluster represents a known class. Most unknown samples locate in the open space, therefore the proposed method is able to distinguish unknown samples from known samples. In the extension phase, the Macro- F_1 is 0.917 and the Micro- F_1 is 0.867. In the retraction phase, the Macro- F_1 is 0.859 and the Micro- F_1 is 0.802. The reason why the Macro- F_1 and Micro- F_1 scores in the extension phase are higher than that in the retraction phase is that, in the extension phase, known-class clusters are much more compact and separable (as shown in Figure 3.12a), while in the retraction phase,

FIGURE 3.12: The feature distribution learned from the trained CNN model using Discriminative Loss ($\lambda = 0.5$) on the pressure testing dataset in the extension phase (a) and retraction phase (b).

FIGURE 3.13: The Macro- F_1 and Micro- F_1 scores experimented on air pressure data with different λ in the extension phase (a) and retraction phase (b).

the variance of each cluster is large and some features from different classes even overlapped each other (as shown in Figure 3.12b), which means features learned in the extension phase could generate more ideal Gaussian distributions to classify known classes and reject unknown samples. More experiments are also conducted to investigate the sensitivity of λ to the performance of the proposed method on the air pressure dataset (see Figure 3.13). The F-measure values (Macro- F_1 and Micro- F_1) of the proposed method on the air pressure dataset remain largely stable across a wide range of λ .

We also experimented on four other classical methods for OSR on the air pressure dataset (results are shown in Table 3.3 and Table 3.4):

1) Softmax: All testing samples are fed into the constructed CNN that is trained using the Softmax loss function. Traditionally Softmax uses maximum probability value for classification. Instead, as hinted in [16], an additional measure is added that if the

FIGURE 3.14: The definition of Area and Interval in the extension phase.

maximum probability value of the Softmax function is less than 0.5, this sample will be labeled as unknown.

2) Openmax: One state-of-the-art deep learning-based OSR method is Openmax [16]. The test is carried out according to [16]. The Weibull tail size is fixed to 10 and the decision threshold is set to 0.5.

3) Mahalanobis Distance: In fault detection, a common method for anomaly detection is Mahalanobis distance [73]. In this experiment, four features are extracted from raw signals: Standard Deviation (STD), Root Mean Square (RMS), Area and Interval (Figure 3.14 is an example to explain the definition of Area and Interval). Thus, each sample is represented by a feature vector $\mathbf{f} = (f_1, f_2, f_3, f_4)^T$. For each class, the mean feature vector $\mathbf{c}_i$, covariance matrix $\mathbf{S}_i$, mean Mahalanobis distance μ_i and its standard deviation σ_i are calculated. For a testing sample x , if $\forall i \in \{1, \dots, K\}, d_M^{(i)}(x) = \sqrt{(\mathbf{f} - \mathbf{c}_i)^T \mathbf{S}_i^{-1} (\mathbf{f} - \mathbf{c}_i)} > \mu_i + 3\sigma_i$ satisfies, this sample will be judged as unknown, otherwise, this sample will be judged as known. For further known classification, a Multi-class Support Vector Machine (MSVM) has been trained with a Radial Basis Function (RBF) kernel [74]. The upper bound of penalty C is set to 128 and the width of the RBF kernel w is set to 1.

4) Eigenface: In the field of images, a classical method for unknown detection is Eigenface [75]. The standard weight vector $\mathbf{\Omega}_i = [\omega_1, \omega_2, \dots, \omega_m]^T$ in each class is obtained by Principal Component Analysis (PCA) [76], where i is the index of class and m is the number of principal components (here m is set to 7). The Euclidean distance ε_i between

TABLE 3.3: Macro- F_1 and Micro- F_1 scores among five methods for the extension phase.

Method	Macro- F_1	Micro- F_1
Eigenface[75]	0.631	0.563
Mahalanobis Distance[73]	0.897	0.854
Softmax[16]	0.711	0.499
Openmax[16]	0.857	0.831
Discriminative Loss	0.917	0.867

TABLE 3.4: Macro- F_1 and Micro- F_1 scores among five methods for the retraction phase.

Method	Macro- F_1	Micro- F_1
Eigenface[75]	0.553	0.526
Mahalanobis Distance[73]	0.808	0.782
Softmax[16]	0.680	0.460
Openmax[16]	0.781	0.625
Discriminative Loss	0.859	0.836

each training sample and its corresponding standard weight vector Ω_i need to be calculated ($\varepsilon_i = \|\Omega_t - \Omega_i\|_2$) to get the mean distance μ_i and its standard deviations σ_i in each class. For a testing sample, if $\forall i \in \{1, \dots, K\}, \varepsilon_i > \mu_i + 3\sigma_i$ satisfies, this sample will be judge as unknown, otherwise, this sample will be judged as known and will be classified as class i where ε_i is minimum.

The Macro- F_1 and Micro- F_1 scores among above four classical methods and the proposed method are listed in Table 3.3 for the extension phase and Table 3.4 for the retraction phase. Compared with other methods, the proposed method achieves the highest Macro- F_1 and Micro- F_1 scores, which means the proposed method is more powerful to classify different levels of air leakage, even unknown air leakage faults. It should be noted that in the retraction phase, among these four classical methods, the best Macro- F_1 score is 0.808 and the best Micro- F_1 score is only 0.782, while the proposed method could increase the Macro- F_1 score by 6% and increase the Micro- F_1 score by 7%. Another point worth noting is that the proposed method is an end-to-end system that uses only a CNN to realize feature extraction, unknown detection, and known classification, while Eigenface [75], Mahalanobis Distance [73], and Openmax [2] are all multi-step methods (see Table 3.5), which means the proposed method is with ease to implement in practical applications.

TABLE 3.5: Comparison of the process in the testing phase among five methods.

Method	Feature Extraction	Unknown Detection	Known Classification
Eigenface[75]	PCA	Euclidean Distance	
Mahalanobis Distance[73]	Manually Design	Mahalanobis Distance	MSVM
Softmax[16]	CNN		
Openmax[16]	CNN		Score Redistribution
Discriminative Loss	CNN		

TABLE 3.6: Comparison of running time (milliseconds per input) among five deep learning-based OSR methods.

Method	Run Time
Softmax[16]	0.179
Openmax[16]	0.181
CROSR[4]	0.184
GDFR[12]	0.772
Discriminative Loss	0.181

3.4.4 Analysis of Efficiency

In practical applications, besides accuracy, efficiency is another important indicator to evaluate system performance. Here, the computational cost during the testing phase among five deep learning-based OSR methods is compared. In this experiment, the network architecture and experimental environment are set to the same. The running time is measured during the testing phase with a batch size equals to 1 and shown in milliseconds per input. From Table 3.6, the running time of the proposed method is only 0.02 milliseconds per input slower than that of Softmax, but much faster than that of CROSR [4] and GDFR [12], which means compared with other deep learning-based OSR methods, the proposed method could raise an alarm more timely when meeting unknown faults.

3.5 Chapter Summary

In this chapter, an online air leakage detection system of a pneumatic train door subsystem using the proposed Discriminative Loss method is developed. In the proposed OSR method, the loss function within Convolutional Neural Networks (CNNs) is modified to force learned features in different classes to be close to different centroids for Gaussian modeling. The proposed method is demonstrated on the MNIST dataset and applied to detect different levels of air leakage on the pneumatic train door subsystem. The pressure data were collected under normal and four different levels of air leakage situations. *Normal*, *leakage 1*, *2*, and *3* these four situations are selected as the known classes which are used during model training and testing, and *leakage 4* is set as the unknown class which is only used during model testing. The proposed method is able to not only classify different levels of air leakage faults but also detect unknown air leakage faults. Compared with Softmax, Openmax, Mahalanobis Distance, and Eigenface, the proposed method achieves higher OSR accuracy, as well as is with ease of implementation. Moreover, the proposed method has a faster running time compared with other deep learning-based OSR methods. More experiments have also been conducted to verify the stability of the proposed method across a wide range of the balancing parameter λ . The developed system provides a feasible and reliable solution for air leakage detection, and it seems to be the potential for the detection of other faults that may appear on the pneumatic train door subsystem. In Chapter 6, Discriminative Loss is also compared with other proposed methods on six standard image distastes.

Chapter 4

Open Set Recognition with Conditional Probabilistic Generative Models

In Chapter 3, an OSR method with a shorter running time is proposed. Although this method achieves promising results on air pressure data, its performance on standard image datasets is not satisfying (results are shown in Chapter 6). To achieve higher accuracy on image datasets, in this chapter, we introduce a novel OSR method, called Conditional Probabilistic Generative Models (CPGM). The core insight of our work is to add discriminative information into the probabilistic generative models (*i.e.*, VAE [14] and AAE [22]), such that the proposed models cannot only detect unknown samples but also classify known classes by forcing different latent features to approximate conditional Gaussian distributions. We discuss many model variants and provide comprehensive experiments to study their characteristics. Experiment results on multiple benchmark datasets reveal that the proposed method significantly outperforms the baselines.

4.1 Introduction

OSR is a challenging problem setting as only data from known classes are provided at training time, which prevents us from training an unknown sample detector in a fully supervised manner. To realize unknown detection, many previous works in the literature model the data distribution of known samples by unsupervised learning methods [17] [18] [19] [20] [21], and those deviated data points from learned distribution are then deemed to be unknown samples. For example, VAE [14] is a popular method

and is often combined with clustering [17], GMM [20], or one-class [18] algorithms, to achieve this goal. VAE is a probabilistic graphical model which is trained not only to reconstruct the input accurately, but also to force the posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ in the latent space to approximate one prior distribution $p_\theta(\mathbf{z})$, such as a multivariate Gaussian or Bernoulli distribution. The well-trained VAE is able to correctly describe the distribution of known data, and deviated samples that locate in low-probability regions will be assumed as unknown. In the same spirit, another line of works relies on generative adversarial networks (GAN) [13] that were proposed, such as AAE [22], to perform variational inference and impose one prior distribution to the data distribution through adversarial training. AAE is able to match the posterior distribution in the latent space with one known prior distribution, such as a multivariate Gaussian distribution. Therefore, an AAE model is commonly trained through unsupervised learning or semi-supervised learning and widely applied to outlier detection [77] [78] [79] [80]. Figure 1.1a and Figure 1.1b provide the visualization of the latent representations of VAE and AAE respectively in low-dimensional space when the prior distribution $p_\theta(\mathbf{z})$ is a multivariate normal Gaussian. As we can see, although these probabilistic models excel at modeling data distributions of known samples for unknown detection, they cannot provide discriminative representations to undertake classification tasks as all data follow a universal distribution.

To overcome this shortcoming, we propose a novel framework, Conditional Probabilistic Generative Models (CPGM), for OSR. Different from previous anomaly detection methods based on probabilistic generative models, the proposed framework is able to generate class conditional posterior distributions $q_\phi(\mathbf{z}|\mathbf{x}, k)$ in the latent space where k is the index of known classes. These conditional distributions are forced to approximate different multivariate Gaussian models $p_\theta^{(k)}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_k, \mathbf{I})$ where $\boldsymbol{\mu}_k$ is the mean of the k -th multivariate Gaussian distribution, obtained by a fully-connected layer that maps the categorical distribution of input to the latent space. Figure 1.1c and Figure 1.1d are examples of latent representations of the proposed method on the MNIST dataset. These learned features will be fed to an open set classifier, which consists of two parts: an unknown detector and a closed set classifier. As known samples tend to follow prior distributions, the unknown detector will recognize those samples locating in lower probability regions as unknown. Meanwhile, for the known sample, the closed set classifier will calculate its prediction scores over all known classes and predict it as the class with the highest score.

Here we mainly focus on two popular probabilistic generative models, namely VAE and AAE. To fully investigate their potentials for the goal of OSR, we explore many model variants and conduct comprehensive experiments to study their characteristics. For example, current networks for classification tend to have deep structures for higher accuracy [81], but traditional VAE models with deep hierarchies are difficult to optimize, due to multiple layers of dependent stochastic variables [82]. We overcome this shortcoming by adopting the probabilistic ladder network [82] in the proposed conditional VAE model, which enables the proposed VAE model to generate an approximate posterior distribution merging information from the bottom-up computed approximate likelihood with top-down prior information.

Through our comprehensive analysis in the experiment, we empirically demonstrate that our framework, with either VAE or AAE as the probabilistic generative model, can significantly outperform baseline methods. Our main contributions are summarized as follows:

- We propose a novel OSR framework, Conditional Probabilistic Generative Models (CPGM). Compared with the original VAE and AAE, the proposed method is able to learn conditional distributions for known classification and unknown detection.
- We explore many model variants of the proposed framework and present much practical guidance to use VAE and AAE for OSR.
- Experiments on several benchmark datasets show that our method brings 9.96% OSR performance gain on the MNIST-based dataset and brings 7.99% OSR performance gain on the CIFAR10-based dataset.

4.2 Preliminary

Before introducing the proposed method, we briefly introduce the terminology and notations of VAE [14] and AAE [22].

4.2.1 VAE

The VAE commonly consists of an encoder, a decoder and a loss function $\mathcal{L}(\theta; \phi; \mathbf{x})$. The encoder is a neural network that has parameters ϕ . Its input is a sample $\mathbf{x}$ and

its output is a hidden representation $\mathbf{z}$. The decoder is another neural network with parameters θ . Its input is the representation $\mathbf{z}$ and it outputs the probability distribution of the sample. The loss function in the VAE is defined as follows:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{\phi}; \mathbf{x}) = & -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{z})) \\ & + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} [\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] \end{aligned} \quad (4.1)$$

where $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is the approximate posterior, $p_{\boldsymbol{\theta}}(\mathbf{z})$ is the prior distribution of the latent representation $\mathbf{z}$ and $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ is the likelihood of the input $\mathbf{x}$ given latent representation $\mathbf{z}$. On the right-hand side of Eqn. (4.1), the first term is the KL-divergence between the approximate posterior and the prior. It can be viewed as a regularizer to encounter the approximate posterior to be close to the prior $p_{\boldsymbol{\theta}}(\mathbf{z})$. The second term can be viewed as the reconstruction errors to force the output to reconstruct the input correctly.

Commonly, the prior over the latent representation $\mathbf{z}$ is the centered isotropic multivariate Gaussian $p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$. In this case, the variational approximate posterior could be a multivariate Gaussian with a diagonal covariance structure:

$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \mathbf{I}) \quad (4.2)$$

where the mean $\boldsymbol{\mu}$ and the standard deviation $\boldsymbol{\sigma}$ of the approximate posterior are outputs of the encoding multi-layered perceptrons (MLPs). The latent representation $\mathbf{z}$ is defined as $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ is the element-wise product. Let J be the dimensionality of $\mathbf{z}$, then the KL-divergence can be calculated [14]:

$$\begin{aligned} & -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{z})) \\ & = \frac{1}{2} \sum_{j=1}^J (1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2) \end{aligned} \quad (4.3)$$

With the loss function $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{\phi}; \mathbf{x})$, the VAE is trained not only to reconstruct the input accurately, but also to force the posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ in the latent space to approximate the prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z})$. If a sample locates in the low probability region of the learned distribution, this sample will be recognized as unknown.

4.2.2 AAE

The AAE [22] model is composed of three modules: an encoder (generator) $G(\mathbf{x})$, a decoder $De(\mathbf{z})$ and a discriminator $D(p(\mathbf{z}))$. The AAE model is trained in two phases: the *reconstruction* phase and the *regularization* phase. In the *reconstruction* phase, the encoder $G(\mathbf{x})$ and decoder $De(\mathbf{z})$ are trained jointly to reconstruct the input $\mathbf{x}$ correctly, which can be expressed as follows:

$$\mathcal{L}_r = \|\mathbf{x} - \tilde{\mathbf{x}}\|_2 \quad (4.4)$$

where $\tilde{\mathbf{x}}$ is the reconstructed image.

In the *regularization* phase, an adversarial game is established between the encoder $G(\mathbf{x})$ and the discriminator $D(p(\mathbf{z}))$: firstly the discriminator $D(p(\mathbf{z}))$ is trained to distinguish the true distribution (generated using the prior distribution $p(\mathbf{z})$) from the posterior distribution $q(\mathbf{z}|\mathbf{x})$ (generated by the encoder $G(\mathbf{x})$), then the encoder $G(\mathbf{x})$ is trained to confuse the discriminator $D(p(\mathbf{z}))$. This adversarial process can be expressed in the following equation:

$$\min_G \max_D E_{p(\mathbf{z})} [\log D(p(\mathbf{z}))] + E_{\mathbf{x} \sim p_{data}} [\log(1 - D(G(\mathbf{x}))) \quad (4.5)$$

When training is completed, the posterior distribution $q(\mathbf{z}|\mathbf{x})$ in the latent space will be close to the prior distribution $p(\mathbf{z})$. If a sample locates in the low probability region of the learned distribution, this sample will be recognized as unknown.

4.3 Method

In this section, we describe the proposed framework based on two probabilistic generative models, *i.e.*, VAE and AAE, which are denoted by CPGM-VAE and CPGM-AAE, respectively. For each method, we first present the architecture of the proposed model and then describe the functions of different modules during training and testing.

4.3.1 CPGM-VAE

4.3.1.1 Architecture

The architecture of CPGM-VAE is composed of four modules (as shown in Figure 4.1):

1. Encoder $\mathcal{F}$
2. Decoder $\mathcal{G}$
3. Known Classifier $\mathcal{C}$
4. Unknown Detector $\mathcal{D}$

FIGURE 4.1: Block diagram of CPGM-VAE: The **encoder $\mathcal{F}$** and **decoder $\mathcal{G}$** are applied with the probabilistic ladder architecture to generate more informative posterior distributions. The **known classifier $\mathcal{C}$** takes latent representations as input and produces the probability distribution over the known classes. The **unknown detector $\mathcal{D}$** is modeled by the conditional Gaussian distributions and reconstruction errors from training samples, which is used for unknown detection. During training, the proposed model is trained to minimize the sum of the reconstruction loss $\mathcal{L}_r$, KL-divergence $\mathcal{L}_{KL}$ (both in the latent space and middle layers) and classification loss $\mathcal{L}_c$. During testing, the **unknown detector $\mathcal{D}$** will judge whether this sampler is unknown by its latent features and reconstruction errors. If this sample is known, the **known classifier $\mathcal{C}$** will give out its predicted label.

Encoder $\mathcal{F}$. To generate more informative posterior distributions, the probabilistic ladder architecture is adopted between the upward path and downward path. In detail, the l -th layer in the encoder $\mathcal{F}$ is expressed as follows:

$$\begin{aligned} x_l &= \text{Conv}(x_{l-1}) \\ h_l &= \text{Flatten}(x_l) \\ \mu_l &= \text{Linear}(h_l) \\ \sigma_l^2 &= \text{Softplus}(\text{Linear}(h_l)) \end{aligned}$$

where Conv is a convolutional layer followed by a batch-norm layer and a Parametric Rectified Linear Unit (PReLU), Flatten is a linear layer to flatten 2-dimensional data into 1-dimension, Linear is a single linear layer and Softplus applies $\log(1 + \exp(\cdot))$ non-linearity to each component of its argument vector (Figure 4.2 illustrates these operations). The latent representation z is defined as $z = \mu + \sigma \odot \epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\odot$ is the element-wise product, and μ, σ are the outputs of the top layer L .

FIGURE 4.2: Operations in the upward pathway of CPGM-VAE.

Decoder $\mathcal{G}$. The l -th layer in the decoder $\mathcal{G}$ is expressed as follows:

$$\begin{aligned} \tilde{c}_{l+1} &= \text{Unflatten}(\tilde{z}_{l+1}) \\ \tilde{x}_{l+1} &= \text{ConvT}(\tilde{c}_{l+1}) \\ \tilde{h}_{l+1} &= \text{Flatten}(\tilde{x}_{l+1}) \\ \tilde{\mu}_l &= \text{Linear}(\tilde{h}_{l+1}) \\ \tilde{\sigma}_l^2 &= \text{Softplus}(\text{Linear}(\tilde{h}_{l+1})) \\ z_l &= \tilde{\mu}_l + \tilde{\sigma}_l^2 \odot \epsilon \end{aligned}$$

where ConvT is a transposed convolutional layer and Unflatten is a linear layer to convert 1-dimensional data into 2-dimension (Figure 4.3 illustrates these operations). In the l -th layer, the bottom-up information (μ_l and σ_l) and top-down information ($\tilde{\mu}_l$ and $\tilde{\sigma}_l$) are interacted by the following equations defined in [82]:

$$q-\mu_l = \frac{\tilde{\mu}_l \tilde{\sigma}_l^{-2} + \mu_l \sigma_l^{-2}}{\tilde{\sigma}_l^{-2} + \sigma_l^{-2}} \quad (4.6)$$

$$q-\sigma_l^2 = \frac{1}{\tilde{\sigma}_l^{-2} + \sigma_l^{-2}} \quad (4.7)$$

FIGURE 4.3: Operations in the downward pathway of CPGM-VAE.

Known Classifier $\mathcal{C}$. The known classifier $\mathcal{C}$ is a Softmax layer, which takes the latent representation z as input. It produces the probability distribution over the known classes.

Unknown Detector $\mathcal{D}$. When training is completed, the unknown detector $\mathcal{D}$ is modeled by information hidden in the latent representations and reconstruction errors. During the testing phase, the unknown detector $\mathcal{D}$ is used as a binary classifier to judge whether the input is known or unknown (details are discussed in Section 4.3.1.3).

4.3.1.2 Training

During the training phase, the proposed method CPGM-VAE forces the conditional posterior distributions $q_\phi(z|x, k)$ to approximate different multivariate Gaussian models $p_\theta^{(k)}(z) = \mathcal{N}(z; \mu_k, \mathbf{I})$ where k is the index of known classes, and the mean of k -th Gaussian distribution μ_k is obtained by a fully-connected layer which maps the one-hot encoding of the input's label to the latent space. The KL-divergence used in latent space (Eqn. (4.3)) is modified as follows:

$$\begin{aligned}
& -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}, k) \parallel p_\theta^{(k)}(\mathbf{z})) \\
& = \int q_\phi(\mathbf{z}|\mathbf{x}, k) (\log p_\theta^{(k)}(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x}, k)) d\mathbf{z} \\
& = \int \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2) (\log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_k, \mathbf{I}) - \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)) d\mathbf{z} \\
& = \frac{1}{2} \sum_{j=1}^J (1 + \log(\sigma_j^2) - (\mu_j - \mu_j^{(k)})^2 - \sigma_j^2)
\end{aligned} \tag{4.8}$$

During the training phase, the model is trained to minimize the sum of the reconstruction loss $\mathcal{L}_r$, KL-divergence $\mathcal{L}_{KL}$ and classification loss $\mathcal{L}_c$. To measure classification loss $\mathcal{L}_c$, we use softmax cross-entropy of prediction and ground-truth labels. The classification loss is defined as

$$\mathcal{L}_c = -\ln S_c \tag{4.9}$$

where S_c is the probability of the target class. To measure reconstruction loss $\mathcal{L}_r$, we use the L_2 distance between input image $\mathbf{x}$ and reconstructed image $\tilde{\mathbf{x}}$. The reconstruction loss $\mathcal{L}_r$ is defined as

$$\mathcal{L}_r = \|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2 \tag{4.10}$$

As the probabilistic ladder architecture is adopted, the KL-divergence is considered not only in the latent space but also in all middle layers:

$$\begin{aligned}
\mathcal{L}_{KL} = & -\frac{1}{L} [D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}, k) \parallel p_\theta^{(k)}(\mathbf{z})) \\
& + \sum_{l=1}^{L-1} D_{KL}(q_\theta(\tilde{\mathbf{x}}_l|\tilde{\mathbf{x}}_{l+1}, \mathbf{x}) \parallel q_\theta(\tilde{\mathbf{x}}_l|\tilde{\mathbf{x}}_{l+1}))]
\end{aligned} \tag{4.11}$$

where

$$q_\theta(\tilde{\mathbf{x}}_l|\tilde{\mathbf{x}}_{l+1}, \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}}_l; \mathbf{q}\boldsymbol{\mu}_l, \mathbf{q}\boldsymbol{\sigma}_l^2) \tag{4.12}$$

$$q_\theta(\tilde{\mathbf{x}}_l|\tilde{\mathbf{x}}_{l+1}) = \mathcal{N}(\tilde{\mathbf{x}}_l; \tilde{\boldsymbol{\mu}}_l, \tilde{\boldsymbol{\sigma}}_l^2) \tag{4.13}$$

The loss function used in our model is summarized as follows:

$$\mathcal{L} = -(\mathcal{L}_r + \beta\mathcal{L}_{KL} + \lambda\mathcal{L}_c) \tag{4.14}$$

where β is increased linearly from 0 to 1 during the training phase as described in [82] and λ is a constant.

4.3.1.3 Testing

When training is completed, we use learned features to model per class multivariate Gaussian models $f_k(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{m}_k, \sigma_k^2)$ where $\mathbf{m}_k$ and σ_k^2 are the mean and variance of the latent representations of all correctly classified training samples in k -th class. If the dimension of the latent space is n : $\mathbf{z} = (z_1, \dots, z_n)$, the probability of a sample locating in the distribution $f_k(\mathbf{z})$ is defined as follows:

$$P_k(\mathbf{z}) = 1 - \int_{m_0 - |z_0 - m_0|}^{m_0 + |z_0 - m_0|} \dots \int_{m_n - |z_n - m_n|}^{m_n + |z_n - m_n|} f_k(\mathbf{t}) d\mathbf{t} \quad (4.15)$$

We also analyze information hidden in the reconstruction errors. The reconstruction errors of input from known classes are commonly smaller than that of unknown classes [11]. Here we obtain the reconstruction error threshold by ensuring 95% training data to be recognized as known. We define the threshold of conditional Gaussian distributions (CGD) As τ_l and the threshold of reconstruction errors (RE) As τ_r . The testing phase can be summarized as follows: for a testing sample X , assuming its latent feature is Z and its reconstruction error is R , if $\forall k \in \{1, \dots, K\}, P_k(\mathbf{Z}) < \tau_l$ **or** $R > \tau_r$, then X will be predicted as unknown, otherwise, X will be predicted as known with label $y_p = \argmax(\mathcal{C}(Z))$. Details of the testing procedure are described in Algo. 3.

4.3.2 CPGM-AAE

4.3.2.1 Architecture

The architecture of the proposed method CPGM-AAE is shown in Figure 4.4a, which is composed of five modules:

1. Encoder (Generator) $\mathcal{F}$
2. Decoder $\mathcal{G}$
3. Discriminator $\mathcal{U}$
4. Known Classifier $\mathcal{C}$
5. Unknown Detector $\mathcal{D}$

To enable AAE to generate discriminative features for known classification, different from traditional VAE models that only extract latent features, we add one more fully-connect layer on the top of the encoder $\mathcal{F}(\mathbf{y}, \mathbf{z}_0|\mathbf{x})$ to extract categorical information of

Algorithm 3: Testing procedure of CPGM-VAE

Require: Testing sample $\mathbf{X}$
Require: Trained modules $\mathcal{F}$, $\mathcal{G}$, $\mathcal{C}$
Require: Threshold τ_l of CGD
Require: Threshold τ_r of RE
Require: For each class k , let $\mathbf{z}_{i,k}$ is the latent representation of each correctly classified training sample $\mathbf{x}_{i,k}$

- 1: **for** $k = 1, \dots, K$ **do**
- 2: compute the mean and variance of each class: $\mathbf{m}_k = \text{mean}_i(\mathbf{z}_{i,k})$,
 $\sigma_k^2 = \text{var}_i(\mathbf{z}_{i,k})$
- 3: model the per class multivariate Gaussian: $f_k(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{m}_k, \sigma_k^2)$
- 4: **end for**
- 5: latent representation $\mathbf{Z} = \mathcal{F}(\mathbf{X})$
- 6: predicted known label $y_{pred} = \text{argmax}(\mathcal{C}(\mathbf{Z}))$
- 7: reconstructed image $\tilde{\mathbf{X}} = \mathcal{G}(\mathbf{Z})$
- 8: reconstruction error $R = \|\mathbf{X} - \tilde{\mathbf{X}}\|_2^2$
- 9: **if** $\forall k \in \{1, \dots, K\}, P_k(\mathbf{Z}) < \tau_l$ **or** $R > \tau_r$ **then**
- 10: predict $\mathbf{X}$ as unknown
- 11: **else**
- 12: predict $\mathbf{X}$ as known with label y_{pred}
- 13: **end if**

input. Therefore, the encoder $\mathcal{F}(\mathbf{y}, \mathbf{z}_0 | \mathbf{x})$ provides not only a latent feature $\mathbf{z}_0$, but also a categorical distribution $\mathbf{y}$ in the top layer, which can be expressed as follows:

$$\begin{aligned} \mathbf{h}_n &= \text{Flatten}(\mathbf{x}_n) \\ \mathbf{z}_0 &= \text{Linear}(\mathbf{h}_n) \\ \mathbf{y} &= \text{Softmax}(\text{Linear}(\mathbf{h}_n)) \end{aligned}$$

where `Softmax` applies the Softmax function to the categorical distribution $\mathbf{y}$ so that the elements of $\mathbf{y}$ lie in the range $[0,1]$ and sum to 1. The latent feature $\mathbf{z}_0$ will be fed to the discriminator $\mathcal{U}$ to approximate a normal Gaussian distribution $\mathcal{N}(\mathbf{z}_0; 0, \mathbf{I})$, while the categorical distribution $\mathbf{y}$ will be fed to the known classifier $\mathcal{C}$ (a Softmax layer) to produce the probability distribution over the known classes for known classification, and a fully-connected layer is used to map $\mathbf{y}$ into the latent space to generate class conditional Gaussian means μ_k where k is the index of known classes. By adding μ_k to the latent feature $\mathbf{z}_0$: $\mathbf{z} = \alpha \times \mu_k + \mathbf{z}_0$ where α is a constant, different latent features $\mathbf{z}$ will be close to different means μ_k to generate class conditional Gaussian distributions $\mathcal{N}_k(\mathbf{z}; \mu_k, \mathbf{I})$. These learned Gaussian models build up the unknown detector $\mathcal{D}$, which will be used during the testing phase to judge whether an input is known or unknown.

FIGURE 4.4: The model architecture of the proposed CPGM-AAE (a) (details of CPGM-AAE are described in Section 4.3.2.1) and two model variants ((b) and (c)) (details of two model variants are described in Section 4.4.4). We visualize the features of samples from known classes generated by these three models in (d), (e) and (f), respectively. All these three models could generate 10 feature clusters and each cluster represents one class. Although AAE-Variant 1 could generate separable features, these features do not follow class conditional Gaussian distributions for unknown detection. Conversely, AAE-Variant 2 could generate ideal class conditional Gaussian distributions, but they are highly overlapped and can hardly provide discriminative information for known classification. In comparison, the proposed method, CPGM-AAE, could generate ideal class conditional Gaussian distributions for unknown detection, and these features are also separable in the feature space for known classification.

We also explore two model variants that are based on AAE, as shown in Figure 4.4b and Figure 4.4c. The main difference between these three models is how to obtain Gaussian means μ_k to generate conditional Gaussian distributions. In AAE-Variant 1 (as shown in Figure 4.4b), similar as our previous model CPGM-VAE, k -th class latent features are forced to approximate k -th Gaussian distribution $\mathcal{N}_k(z; \mu_k, \mathbf{I})$ where Gaussian means μ_k are obtained from a separate fully-connected layer that maps the one-hot encoding of the input's label to the latent space. While in AAE-Variant 2 (as shown in Figure 4.4c), Gaussian means μ_k are achieved by mapping the learned categorical distribution y into the latent space through a fully-connected layer. Details of comparison between CPGM-AAE, AAE-Variant 1 and AAE-Variant 2 are discussed in Section 4.4.4.

4.3.2.2 Training

There are four phases during the CPGM-AAE training procedure: the *reconstruction* phase, the *regularization* phase, the *classification* phase and the *center-learning* phase.

In the *reconstruction* phase, the encoder $\mathcal{F}$ and decoder $\mathcal{G}$ are trained jointly to minimize the reconstruction loss $\mathcal{L}_r$: the L_2 distance between the input image $\mathbf{x}$ and the reconstructed image $\tilde{\mathbf{x}}$, which is defined as follows:

$$\mathcal{L}_r = \|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2 \quad (4.16)$$

In the *regularization* phase, firstly the discriminator $\mathcal{U}$ is trained to distinguish the prior distribution $p(\mathbf{z})$ from the posterior distribution $q(\mathbf{z}|\mathbf{x}) = \mathcal{F}(\mathbf{x})$, then the encoder $\mathcal{F}$ is trained to confuse the discriminator $\mathcal{U}$. This adversarial process can be expressed in the following equation:

$$\min_{\mathcal{F}} \max_{\mathcal{U}} E_{p(\mathbf{z})}[\log \mathcal{U}(p(\mathbf{z}))] + E_{\mathbf{x} \sim p_{data}}[\log(1 - \mathcal{U}(\mathcal{F}(\mathbf{x})))] \quad (4.17)$$

In the *classification* phase, the known classifier $\mathcal{C}$ is trained to minimize the classification loss $\mathcal{L}_c$ between predicted scores and ground-truth labels, which is defined as follows:

$$\mathcal{L}_c = -\ln S_c \quad (4.18)$$

where S_c is the probability of the target class.

In the *center-learning* phase, the mean of the k -th multivariate Gaussian distribution $\boldsymbol{\mu}_k$ is achieved by mapping the categorical distribution $\mathbf{y}$ into the latent space. Meanwhile, to keep the different distributions as far away as possible, we add a distance loss $\mathcal{L}_d$ to linearly penalize the Euclidean distance between every two centers. Specifically, if the Euclidean distance is larger than a threshold η , the distance loss $\mathcal{L}_d$ will be zero. This distance loss $\mathcal{L}_d$ can be expressed as the following equation:

$$\mathcal{L}_d(i, j) = \begin{cases} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2^2 & \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2^2 \leq \eta \\ 0 & \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2^2 > \eta \end{cases} \quad (4.19)$$

4.3.2.3 Testing

As latent features are also forced to approximate class conditional Gaussian distributions, the testing phase of CPGM-AAE is the same as CPGM-VAE, which has been described in Section 4.3.1.3 and summarized in Algo. 3.

4.4 Experiments

To evaluate the performance of the proposed methods for OSR, we conduct extensive experiments on multiple standard image datasets. In this section, firstly we present important implementation details and introduce used datasets in our experiments. Then, we conduct various ablative experiments to validate each component in the proposed CPGM-VAE and CPGM-AAE. We also compare CPGM-AAE with two AAE-based variants from a qualitative analysis and a quantitative analysis. Then we compare proposed methods with state-of-the-art methods on popular benchmark datasets. Finally, we conduct a case study to investigate and compare the characteristics between the proposed CPGM-VAE and CPGM-AAE for OSR.

4.4.1 Implementation Details

For a fair comparison with previous works, we employ a re-designed VGGNet defined in [4] as our model backbone, which is widely used in the open set recognition literature. The batch size is fixed to 64 and the dimensionality of the latent representation z is fixed to 32. The networks were trained without any large degradation in closed set accuracy from the original ones. The closed set accuracies of the networks for each dataset are listed in Table 4.1. In OSR experiments, the threshold τ_r of reconstruction errors is obtained by ensuring 95% training data be recognized as known, and the threshold of conditional Gaussian distributions τ_l is set to 0.5.

Especially, for CPGM-VAE, we use the SGD optimizer with a learning rate of 0.001, and the parameter β in the loss function (Eqn. (4.14)) is increased linearly from 0 to 1 during the training phase as described in [82], while the parameter λ in Eqn. (4.14) is set equal to 100. For CPGM-AAE, we use the SGD optimizer with a learning rate of 0.1. The constant α (described in Section 4.3.2.1) and the Euclidean distance threshold η in

TABLE 4.1: Comparison of closed set testing accuracies between the plain CNN and the proposed CPGM methods. Although the training objective of CPGM is classifying known samples as well as learning conditional Gaussian distributions, there is no significant degradation in closed set accuracy.

Architecture	MNIST	SVHN	CIFAR-10
Plain CNN	0.997	0.944	0.912
CPGM-VAE	0.996	0.942	0.912
CPGM-AAE	0.995	0.937	0.906

Eqn. (4.19) are respectively set equal to 10 and 1. The dimensionality of the categorical distribution $\mathbf{y}$ equals the number of known classes.

4.4.2 Dataset Description

We conduct OSR experiments on seven popular benchmark datasets: MNIST [57], CIFAR-10 [3], CIFAR-100 [83], ImageNet [84] and LSUN [85].

MNIST. MNIST [57] is the most popular hand-written digit benchmark. There are 60,000 images for training and 10,000 images for testing from ten digit classes. Each image is monochrome with a size of 28×28 . Although near 100% accuracy has been achieved in the closed set classification [86], the open set extension of MNIST still remains a challenge due to the variety of possible outliers.

CIFAR-10. The CIFAR-10 dataset [3] consists of 5,000 training images and 1,000 testing images in each of the ten natural classes. Each image is colorful with a size of 32×32 . In CIFAR-10, each class has large intra-class diversities by color, style, or pose difference.

CIFAR-100. The CIFAR-100 dataset [83] is just like the CIFAR-10 dataset, except it has 100 natural classes containing 600 images each. There are 500 training images and 100 testing images in each class.

ImageNet. ImageNet [84] is a dataset of over 15 million labeled high-resolution images belonging to roughly 22,000 categories. The images were collected from the web and labeled by human labor.

LSUN. The LSUN dataset [85] contains around 10 million labeled images in 10 scene categories (around one million labeled images in each category) and 59 million labeled images in 20 object categories.

4.4.3 Ablation Study of CPGM-VAE

In this section, we analyze our contributions from each component of the proposed CPGM-VAE on the CIFAR-100 dataset [83]. The CIFAR-100 dataset consists of 100 classes, containing 500 training images and 100 testing images in each class. For ablation analysis, the performance is measured by F-measure (or F_1 -scores) [72] against varying Openness [2]. Openness is defined as follows:

$$Openness = 1 - \sqrt{\frac{2 \times N_{train}}{N_{test} + N_{target}}} \quad (4.20)$$

where N_{train} is the number of known classes seen during training, N_{test} is the number of classes that will be observed during testing, and N_{target} is the number of classes to be recognized during testing. We randomly sample 15 classes out of 100 classes as known classes and varying the number of unknown classes from 15 to 85, which means Openness is varied from 18% to 49%. The performance is evaluated by the macro-average F_1 -scores in 16 classes (15 known classes and *unknown*).

We compare the following baselines for ablation analysis of CPGM-VAE:

I. CNN: In this baseline, only the encoder $\mathcal{F}$ (without ladder architecture) and the known classifier $\mathcal{C}$ are trained for closed set classification. This model can be viewed as a plain convolutional neural network (CNN). During testing, learned features will be fed to the known classifier $\mathcal{C}$ to get the probability scores of known classes. A sample will be recognized as unknown if its maximum probability score of known classes is less than 0.5.

II. CVAE: The encoder $\mathcal{F}$, decoder $\mathcal{G}$ and classifier $\mathcal{C}$ are trained without the ladder architecture, and the testing procedure is the same as baseline I. This model can be viewed as a class conditional Variational Autoencoder (CVAE).

III. LCVAE: The probabilistic ladder architecture is adopted in the CVAE, which contributes to the KL-divergences during training (Eqn. (4.11)). We call this model LCVAE. The testing procedure is the same as the baseline I and II.

IV. CVAE+CGD: In this baseline, the model architecture and the training procedure are the same as baseline II, while the conditional Gaussian distributions (CGD) are used to

detect unknown samples during testing.

V. LCVAE+CGD: In this baseline, LCVAE is introduced along with a CGD-based unknown detector. The training and testing procedure are respectively the same as baseline III and IV.

VI. LCVAE+RE: In this baseline, different from baseline V, reconstruction errors (RE), instead of CGD, are used in LCVAE to detect unknown samples.

VII. CPGM-VAE (Proposed Method): In this baseline, the training procedure is the same as baseline V and VI, while during testing, CGD and RE are jointly used for unknown detection.

FIGURE 4.5: F_1 -scores against varying Openness with different baselines (described in Section 4.4.3) for the ablation study of CPGM-VAE. CPGM-VAE achieves the best F_1 -scores by using conditional Gaussian distributions (CGD) and reconstruction errors (RE) jointly to detect unknown samples with the probabilistic ladder architecture.

The experimental results are shown in Figure 4.5. Among baseline I (CNN), II (CVAE), and III (LCVAE), unknown detection simply relies on the known classifier $\mathcal{C}$, there is no significant improvement in OSR performance. Although the OSR performance is a little improved when the probabilistic ladder architecture is adopted (baseline III: LCVAE), the overall performance in these three baselines is weak as the F_1 -scores degrade rapidly as the Openness increases. Conditional Gaussian distributions (CGD) are used for unknown detection in the CVAE model (baseline IV: CVAE+CGD), but it has seen no visible change in performance. In baseline V (LCVAE+CGD), this trend is alleviated by introducing a CGD-based unknown detector into LCVAE. This shows the importance of the probabilistic ladder architecture for OSR. It is also the reason why the CGD-based

unknown detection achieves better performance in baseline V (LCVAE+CGD) than in baseline IV (CVAE+CGD). If we only use reconstruction errors (RE) to detect unknown samples (baseline VI: LCVAE+RE), the results are worst. However, if RE and CGD are jointly used in the unknown detector (baseline VII: CPGM-VAE), there is a little performance improvement. As a result, applying conditional Gaussian distributions (CGD) and reconstruction errors (RE) jointly to detect unknown samples with the probabilistic ladder architecture achieves the best performance (baseline VII: CPGM-VAE).

4.4.4 Model Analysis of CPGM-AAE

In this section, we compare the characteristics between the proposed CPGM-AAE and two model variants that are based on AAE, as shown in Figure 4.4. The architectures of these three models are described as follows:

- CPGM-AAE: The architecture of CPGM-AAE is shown in Figure 4.4a and has been described in Section 4.3.2.1.
- AAE-Variant 1: The architecture of AAE-Variant 1 is shown in Figure 4.4b. Similar to our previous work (CPGM-VAE), k -th class latent features are forced to approximate k -th Gaussian distribution $\mathcal{N}_k(\mathbf{z}; \boldsymbol{\mu}_k, \mathbf{I})$ in the regularization phase, where Gaussian means $\boldsymbol{\mu}_k$ are obtained from a separate fully-connected layer that maps the one-hot encoding of the input's label to the latent space. Inspired from the CGAN model [87], one-hot encoding information is concatenated to latent features, then they are fed into the discriminator $\mathcal{U}$ for discriminator training. The latent feature $\mathbf{z}$ is also fed to the known classifier $\mathcal{C}$ for known classification and the unknown detector $\mathcal{D}$ for unknown detection, respectively.
- AAE-Variant 2: The architecture of AAE-Variant 2 is shown in Figure 4.4c. Different from AAE-Variant 1, the encoder $\mathcal{F}$ is trained not only to obtain the latent feature $\mathbf{z}$ but also to get the categorical distribution $\mathbf{y}$. Gaussian means $\boldsymbol{\mu}_k$ are achieved by mapping the learned categorical distribution $\mathbf{y}$ into the latent space through a fully-connected layer. Meanwhile, $\mathbf{y}$ is fed to the known classifier $\mathcal{C}$ for known classification, and $\mathbf{z}$ is fed to the unknown detector $\mathcal{D}$ for unknown detection.

To evaluate the performance of these three AAE-based models on OSR, we draw 2-dimensional feature visualizations for qualitative analysis and calculate F_1 -scores against varying Openness [2] for quantitative analysis.

Qualitative Analysis. For qualitative analysis, we conduct experiments on the MNIST dataset on these three models and draw their latent features on 2-dimensional space respectively for visualization, as shown in Figure 4.4d, Figure 4.4e, and Figure 4.4f. Through these 2-dimensional latent features, we find all these three models could generate 10 feature clusters and each cluster represents one class. Although AAE-Variant 1 could generate separable features, these features do not follow class conditional Gaussian distributions for unknown detection. Conversely, AAE-Variant 2 could generate ideal class conditional Gaussian distributions, but they are highly overlapped and can hardly provide discriminative information for known classification. In comparison, the proposed method, CPGM-AAE, could generate ideal class conditional Gaussian distributions for unknown detection, and these features are also separable in the feature space for known classification.

Quantitative Analysis. We also conduct experiments on these three models for quantitative analysis by calculating F_1 -scores against varying Openness. Here we still use the MNIST dataset (10 digit classes) as known classes and use 52 letter classes in the EMNIST dataset [88] as unknown classes. Performance is evaluated by macro-average F_1 -scores [72] against varying Openness [2] from 18% to 47% by randomly sampling 10, 14, 19, 25, 32, 42, 52 classes respectively from unknown classes. The experimental results are shown in Figure 4.6. As the Openness increases, F_1 -scores of these three models all decrease, but CPGM-AAE always achieves the best performance, which is consistent with the results in a qualitative analysis.

Moreover, we conduct ablative experiments to explore contributions from each component of CPGM-AAE. We compare the following baselines:

I. CNN: In this baseline, only the encoder (generator) $\mathcal{F}$ and the known classifier $\mathcal{C}$ are trained for closed set classification. This model can be viewed as a plain convolutional neural network (CNN). During testing, learned features will be fed to the known classifier $\mathcal{C}$ to get the probability scores of known classes. A sample will be recognized as unknown if its maximum probability score of known classes is less than 0.5.

II. CAAE: The encoder (generator) $\mathcal{F}$, decoder $\mathcal{G}$, discriminator $\mathcal{U}$ and known classifier $\mathcal{C}$ are trained for closed set classification. The testing procedure is the same as baseline I. This model can be viewed as a class conditional Adversarial Autoencoder (CAAE).

FIGURE 4.6: F_1 -scores against varying Openness with three AAE-based models (described in Section 4.4.4). Among these three models, CPGM-AAE always achieves the best F_1 -scores under varying Openness.

III. CAAE+CGD: In this baseline, the model architecture and the training procedure are the same as baseline II, while the conditional Gaussian distributions (CGD) are used to detect unknown samples during testing.

IV. CAAE+RE: Different from baseline III, reconstruction errors (RE), instead of CGD, are used in CAAE to detect unknown samples.

V. CPGM-AAE (Proposed Method): In this baseline, the training procedure is the same as baseline III and IV, while during testing, CGD and RE are used jointly for unknown detection.

The experimental results are shown in Figure 4.7. From the baseline I (CNN) and II (CAAE), unknown detection simply relies on the known classifier $\mathcal{C}$, there is no obvious change in OSR performance when adding the reconstruction loss function (baseline II: CAAE) into the training procedure of a plain CNN (baseline I: CNN). However, the performance is a little improved when condition Gaussian distributions (CGD) are used for unknown detection (baseline III: CAAE+CGD). If we only use reconstruction errors (RE) to detect unknown samples (baseline IV: CAAE+RE), the performance has been greatly improved, which has exactly the reverse effect in CPGM-VAE as shown by ‘LCVAE+RE’ in Figure 4.5 (more analysis will be discussed in Section 4.4.6). Baseline V (CPGM-AAE) achieves the best performance where RE and CGD are used jointly for unknown detection.

FIGURE 4.7: F_1 -scores against varying Openness with different baselines (described in Section 4.4.4) for the ablation study of CPGM-AAE. CPGM-AAE achieves the best F_1 -scores by using conditional Gaussian distributions (CGD) and reconstruction errors (RE) jointly to detect unknown samples.

4.4.5 Comparison with State-of-the-art Results

In this section, we compare the proposed method with state-of-the-art methods on OSR tasks. Performance is measured by macro-averaged F_1 -scores in all known classes and *unknown*. In the following experiments, the models are trained by all known classes, but in the testing phase, samples from unknown classes are added to the test set.

FIGURE 4.8: Examples from MNIST, Omniglot, MNIST-NoiSe, and Noise datasets.

Firstly, we choose MNIST, the most popular hand-written digit dataset, as the training set. As outliers, we follow the set up in [4], using Omniglot [89], MNIST-NoiSe, and Noise these three datasets. Omniglot is a dataset containing various alphabet characters. Noise is a synthesized dataset by setting each pixel value independently from a uniform distribution on $[0, 1]$. MNIST-NoiSe is also a synthesized dataset by adding noise on MNIST testing samples. Each dataset contains 10, 000 testing samples, the same as

TABLE 4.2: Open set classification results on the MNIST dataset with various outliers added to the test set as unknowns. The performance is evaluated by macro-averaged F_1 -scores in 11 classes (10 known classes and *unknown*). Our methods significantly outperform the baseline methods.

Method	Omniglot	MNIST-noise	Noise
Softmax	0.595	0.801	0.829
Openmax [16]	0.780	0.816	0.826
CROSR [4]	0.793	0.827	0.826
CPGM-VAE	0.850	0.887	0.859
CPGM-AAE	0.872	0.865	0.872

TABLE 4.3: Open set classification results on the CIFAR-10 dataset with various outliers added to the test set as unknowns. The performance is evaluated by macro-averaged F_1 -scores in 11 classes (10 known classes and *unknown*). Our methods achieve the best performance among baseline methods.

*We report the experimental results reproduced in [4].

Method	ImageNet crop	ImageNet resize	LSUN crop	LSUN resize
Softmax [4]*	0.639	0.653	0.642	0.647
Openmax [16]	0.660	0.684	0.657	0.668
LadderNet+Softmax [4]	0.640	0.646	0.644	0.647
LadderNet+Openmax [4]	0.653	0.670	0.652	0.659
DHRNet+Softmax [4]	0.645	0.649	0.650	0.649
DHRNet+Openmax [4]	0.655	0.675	0.656	0.664
CROSR [4]	0.721	0.735	0.720	0.749
C2AE [11]	0.837	0.826	0.783	0.801
CPGM-VAE	0.840	0.832	0.806	0.812
CPGM-AAE	0.793	0.830	0.820	0.865

MNIST, and this makes the known-to-unknown ratio 1:1. Figure 4.8 shows examples of these datasets. The open set recognition scores are shown in Table 4.2 and the proposed method achieves the best results on all given datasets. It also should be noted that CPGM-AAE is more effective on Omniglot and Noise these two datasets, but not as good as CPGM-VAE on the MNIST-Noise dataset.

Secondly, following the protocol defined in [4], all samples in the CIFAR-10 dataset are collected as known data, and samples from other datasets, i.e., ImageNet [84] and LSUN [85], are selected as unknown samples. We resize or crop the unknown samples to make them have the same size as known samples. ImageNet-crop, ImageNet-resize,

FIGURE 4.9: tSNE [1] visualization of feature distributions of testing samples in CPGM-VAE (a) and CPGM-AAE (b). In both methods, samples from different known classes distribute in different clusters while unknown samples are mostly located in open space. The comparison between CPGM-VAE and CPGM-AAE shows that the variance of each cluster is large in CPGM-VAE and the clusters of known classes are more compact in CPGM-AAE. Therefore, when using conditional Gaussian distributions for OSR, CPGM-AAE yields better results.

LSUN-crop, and LSUN-resize these four datasets are generated, and each dataset contains 10,000 testing images through random selection, which is the same as CIFAR-10. This makes during testing the known-to-unknown ratio is 1:1. The performance is evaluated by macro-averaged F_1 -scores in 11 classes (10 known classes and *unknown*), and results are shown in Table 4.3. We can see from the results that on all given datasets, the proposed method is more effective than previous methods and achieves a new state-of-the-art performance. It should be noted that CPGM-VAE yields better OSR performance compared to CPGM-AAE on the ImageNet dataset. Conversely, CPGM-AAE is more effective than CPGM-VAE on the LSUN dataset.

4.4.6 CPGM-VAE vs. CPGM-AAE

We conduct a case study on the CIFAR10 dataset where all animal categories (bird, cat, deer, dog, frog, and horse) are selected as known classes, while all vehicle categories (airplane, car, ship, and truck) are considered to be unknown classes. We compare the performance between the proposed CPGM-VAE and CPGM-AAE by the macro-averaged F_1 -scores in known classes and *unknown*. CPGM-VAE produces an F_1 -scores of 0.655 and CPGM-AAE produces an F_1 -scores of 0.697 (results shown in Table 6.4 in Chapter 6).

FIGURE 4.10: Reconstruction error histograms of known and unknown classes in CPGM-VAE (a) and CPGM-AAE (b). CPGM-AAE has a more clear decision boundary between known and unknown classes, which explains why additionally using reconstruction errors to detect unknown samples at test time can significantly improve performance on CPGM-AAE (the comparison between ‘CAAE+CGD’ and ‘CPGM-AAE (proposed)’ as shown in Figure 4.7) but no obvious improvement on CPGM-VAE (the comparison between ‘LCVAE+CGD’ and ‘CPGM-VAE (proposed)’ as shown in Figure 4.5).

To investigate why CPGM-AAE achieves a better performance in this task, we use tSNE [1] to draw the final feature distributions of testing samples on a 2-dimensional space for both CPGM-VAE and CPGM-AAE, as shown in Figure 4.9. In both cases, known-class features are clustered into six separate clusters and each cluster represents a known class. Most unknown samples are located in open space. However, we note that known clusters appearing under CPGM-AAE (Figure 4.9b) are more compact compared to CPGM-VAE (Figure 4.9a). Therefore, when using conditional Gaussian distributions for OSR, CPGM-AAE yields better results.

Figure 4.10 visualizes reconstruction error histograms generated from known (shown in green) and unknown (shown in yellow) samples in CPGM-VAE and CPGM-AAE. As evident from Figure 4.10, CPGM-AAE has a more clear decision boundary between known and unknown classes than CPGM-VAE. This explains why at the test time, additional using reconstruction errors can significantly improve CPGM-AAE performance (the comparison between ‘CAAE+CGD’ and ‘CPGM-AAE (proposed)’ as shown in Figure 4.7), while in CPGM-VAE, additional using reconstruction errors only achieves a little improvement (the comparison between ‘LCVAE+CGD’ and ‘CPGM-VAE (proposed)’ as shown in Figure 4.5). A possible reason why CPGM-AAE could generate a more clear decision boundary on reconstruction errors between known and unknown classes is that, like most GAN-based models, the training procedure of CPGM-AAE is divided into several phases and one of these phases is dedicated to reducing reconstruction errors. However, like most VAE-based models, CPGM-VAE is trained not

FIGURE 4.11: Reconstructed images generated from CPGM-AAE trained on known classes. All reconstructed images take the form of a blurry version of the input images, but details are preserved better in reconstructions of known samples. Predicting labels from reconstructions of unknown samples is difficult. This model seems to generate a ‘cat’ from a ship and generate a ‘dog’ from a truck (marked in red borders).

only to reconstruct the input accurately but also to reduce the KL-divergence loss and classification loss at the same phase, which may weaken the ability of VAE models to reconstruct inputs accurately. In Figure 4.11, we visualize reconstructed images (of randomly chosen samples) obtained through CPGM-AAE. According to Figure 4.11, all reconstructed images take the form of a blurry version of the input images. However, it should be noted that reconstructions of known samples carry more details compared to reconstructions of unknown samples. Moreover, it is hard to predict the label of an unknown sample by merely looking at its reconstructed image. For example, this model seems to generate a ‘cat’ from a ship and generate a ‘dog’ from a truck (marked in red borders in Figure 4.11).

4.5 Chapter Summary

In this chapter, we have presented a novel framework, Conditional Probabilistic Generative Models (CPGM), for OSR. We focus on two probabilistic generative models, namely VAE and AAE. Compared with previous methods solely based on VAEs or AAEs, the proposed method cannot only classify known samples but also detect unknown samples by forcing posterior distributions in the latent space to approximate different Gaussian models. For CPGM-VAE, the probabilistic ladder architecture is adopted to preserve the information that may vanish in the middle layers. This ladder architecture obviously improves the OSR performance. For CPGM-AAE, three model variants of AAE are explored from a qualitative analysis and a quantitative analysis to investigate their characteristics for OSR. Experiments on several standard image datasets show that the proposed methods significantly outperform the baseline methods. A case study between

CPGM-VAE and CPGM-AAE is also conducted to investigate and compare their characteristics: CPGM-AAE could generate more compact known clusters for OSR and have a more clear decision boundary between known and unknown classes.

Chapter 5

Maximal Mutual Information Open Set Recognition

In Chapter 4, we use AE [90] models to generate latent features with desired statistical characteristics for OSR. Meanwhile, AE models also can extract class-specific features through their reconstruction training strategy. Class-specific features are beneficial for open set classifiers to categorize known classes as well as identify unknown samples [12]. This reconstruction strategy requires the network to restore the input image on pixel-level. This strategy is commonly over-demanding for OSR since class-specific features are generally contained in target objects, not in all pixels. To address this shortcoming, here we discard the pixel-level reconstruction strategy and pay more attention to improving the effectiveness of class-specific feature extraction. We propose a mutual information (MI) [23]-based method with a streamlined architecture, Maximal Mutual Information Open Set Recognition (M2IOSR). The proposed M2IOSR only uses an encoder to extract class-specific features by maximizing the MI between the given input and its latent features across multiple scales. Meanwhile, to further reduce the open space risk, latent features are constrained to class conditional Gaussian distributions by a KL-divergence loss function. In this way, a strong function is learned to prevent the network from mapping different observations to similar latent features and help the network extract class-specific features with desired statistical characteristics. The proposed approach M2IOSR significantly improves the performance of baselines and achieves new state-of-the-art results on several benchmarks consistently.

5.1 Introduction

To train an effective OSR classifier, the features fed into the classifier should be specific enough since only samples from known classes are available at training time. With class-specific features, the classifier can categorize known classes as well as identify unknown samples during testing [12]. Many recent OSR methods [12] [4] [11] rely on AE models to extract class-specific features. AE models are well known to extract class-specific features through a pixel-level reconstruction strategy, where each pixel in the reconstructed image is expected to be exactly the same as the input image. It is usually resource-consuming due to the decoder and complex network architecture. Besides, as shown in Figure 1.2, the class activation maps indicate that an AE model weakens the feature saliency of target objects and only locates the target object well when the proportion of background is small. These limitations have rendered that such a pixel-level reconstruction strategy is less suitable for OSR tasks. As class-specific features are generally contained in target objects, not in the background, there is no need to extract features by pixel-level. It is experimentally observed in Figure 1.2 that a non-pixel-level model is more accurate in class-specific feature extraction and robust in target localization than a pixel-level reconstruction model (see Section 5.4.4 for experiment details).

Considering the above discussions, we discard the pixel-level reconstruction strategy and pay more attention to improving the effectiveness of class-specific feature extraction. A novel OSR method, Maximal Mutual Information Open Set Recognition (M2IOSR), is thus proposed in this work. Different from most previous deep-learning-based OSR methods [9] [10] [4] [11] [12], M2IOSR has a streamlined architecture and only uses an encoder to discover class-specific features by maximizing the mutual information (MI) [23] between the input and its latent features. The mutual information greatly helps the encoder establish stronger interdependence between the input image and its learned features for OSR.

Following the set up in [24], each training iteration of the proposed method includes two steps: maximizing MI on positive pairs from the same input and minimizing MI on negative pairs from different inputs. For positive/negative pairs, the proposed method maximizes/minimizes the MI between the complete input and latent features (global MI), as well as the average MI between every local region (patches) of the input and latent features (local MI). Meanwhile, to further reduce the open space risk [2], it also

encourages latent features to have desired statistical characteristics. For statistical constraint, we replace the adversarial learning in [24] by the KL-divergence loss function proposed in Chapter 4. At the same time, inspired by [53], to prevent MI from being lost in the middle layers, we also maximize/minimize the local MI across multiple scales. In this way, a strong function can be learned to avoid the network from mapping different observations to similar latent features and help the network extract class-specific features with class conditional Gaussian distributions.

In experiments, we carefully study and demonstrate the effectiveness of MI maximization strategy, KL-divergence loss function, multi-scaled local MI, and positive/negative pairs, on improving OSR performance. The contributions of this paper are listed in the following:

- We propose a novel OSR method, Maximal Mutual Information Open Set Recognition (M2IOSR). Different from most previous deep learning-based OSR methods, the proposed method only uses an encoder architecture to extract class-specific features for OSR.
- To further reduce the open space risk, we adopt a KL-divergence loss function to encourage latent features to follow class conditional Gaussian distributions.
- To prevent MI from being lost in the middle layers, we incorporate multi-scaled local MI maximization.
- Experiments on several benchmarks demonstrate that the proposed method significantly enhances the performance of our baseline and achieves new state-of-the-art OSR performance.

5.2 Preliminaries

Before describing our proposed approach, we here briefly revisit the theory of mutual information [23].

In information theory, the Mutual Information (MI) $I(A; B)$ is a measure of the mutual dependence between the two variables A and B . More specifically, it quantifies the

“amount of information” about the variable A (or B) captured via observing the other variable B (or A). The MI can be expressed as the difference between two entropy terms:

$$I(A; B) = H(A) - H(A|B) = H(B) - H(B|A), \quad (5.1)$$

where $H(A)$ and $H(B)$ are the marginal entropy of A and B . This definition has an intuitive interpretation: $I(A; B)$ is the reduction of uncertainty in one variable when the other variable is observed.

If A and B are independent to each other, their MI is zero, *i.e.* $I(A; B) = 0$. In this case, knowing A does not give any information about B and vice versa. At the other extreme, if A and B are interdependent by a deterministic function, then all information conveyed by A is shared with B . Thus, knowing A determines the value of B and vice versa, where the maximum MI is attained.

In the case of jointly continuous random variables, the MI can be expressed as the following formulation:

$$\begin{aligned} I(A; B) &= \int \int p(a, b) \log \frac{p(a, b)}{p(a)p(b)} da db \\ &= \int \int p(b|a)p(a) \log \frac{p(b|a)}{p(b)} da db, \end{aligned} \quad (5.2)$$

where $p(a, b)$ is the joint probability density function of A and B , and $p(a)$ and $p(b)$ are the marginal probability density functions of A and B respectively.

5.3 Proposed Method

In this section, firstly, we introduce the derivation of the loss function used in the proposed approach, Maximal Mutual Information Open Set Recognition (M2IOSR). Then we demonstrate the details of our MI maximization strategy. Finally, we present our training and testing procedures.

5.3.1 Loss Function Derivation

The MI $I(\mathbf{X}; \mathbf{Z})$ measures how much the mutual dependence between the input image $\mathbf{X}$ and the corresponding learned feature $\mathbf{Z}$. Maximizing MI $I(\mathbf{X}; \mathbf{Z})$ ensures the learned feature $\mathbf{Z}$ is highly semantic and representative to describe the input image, which is desired for detecting unknown samples and assigning correct labels to known samples [12]. As introduced in Section 5.2, the MI $I(\mathbf{X}; \mathbf{Z})$ can be expressed as the following formulation:

$$I(\mathbf{X}; \mathbf{Z}) = \int \int p(\mathbf{z}|\mathbf{x})p(\mathbf{x}) \log \frac{p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} d\mathbf{x}d\mathbf{z}, \quad (5.3)$$

where $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z}|\mathbf{x})p(\mathbf{x})$ is the joint probability density function of $\mathbf{X}$ and $\mathbf{Z}$, $p(\mathbf{x})$ and $p(\mathbf{z})$ are the marginal probability density functions of $\mathbf{X}$ and $\mathbf{Z}$ respectively.

To further reduce open space risk [2], latent features should be constrained to some statistical property (*e.g.*, Gaussian distribution). Different from [24] that employs adversarial learning, here we adopt a KL-divergence function to obtain desired latent features. Our training objective is to: 1) maximize MI $I(\mathbf{X}; \mathbf{Z})$ between the input $\mathbf{X}$ and the learned feature $\mathbf{Z}$; and 2) minimize the KL-divergence between the posterior distribution $p(\mathbf{z})$ and the prior distribution $q(\mathbf{z})$. The loss function used in our M2IOSR can be expressed in the following:

$$L = -L_{MI} + \gamma L_{KL}, \quad (5.4)$$

where γ is the balancing parameter, and L_{MI} and L_{KL} respectively denotes the MI loss and the KL-divergence loss. The general form of MI is expressed in Eqn. (5.3), and the general form of KL-divergence function is

$$D_{KL} [p(\mathbf{z})||q(\mathbf{z})] = \int p(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z}. \quad (5.5)$$

Herein, the training objective in Eqn. (5.4) can be expressed as the following formulation:

$$\min_{p(\mathbf{z}|\mathbf{x})} \left\{ - \int \int p(\mathbf{z}|\mathbf{x})p(\mathbf{x}) \log \frac{p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} d\mathbf{x}d\mathbf{z} + \gamma \int p(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z} \right\}. \quad (5.6)$$

As $p(\mathbf{z}) = \int p(\mathbf{z}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$, Eqn. (5.6) is converted to:

$$\begin{aligned}
& \min_{p(\mathbf{z}|\mathbf{x})} \left\{ \int \int p(\mathbf{z}|\mathbf{x})p(\mathbf{x}) \left[-(1+\gamma) \log \frac{p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} \right. \right. \\
& \quad \left. \left. + \gamma \log \frac{p(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})} \right] d\mathbf{x}d\mathbf{z} \right\} \\
& = \min_{p(\mathbf{z}|\mathbf{x})} \left\{ -(1+\gamma) \left[D_{KL}(p(\mathbf{z}|\mathbf{x})p(\mathbf{x})||p(\mathbf{z})p(\mathbf{x})) \right] \right. \\
& \quad \left. + \gamma \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \left[D_{KL}(p(\mathbf{z}|\mathbf{x})||q(\mathbf{z})) \right] \right\}.
\end{aligned} \tag{5.7}$$

where $\mathbb{E}[*]$ denotes the mathematical expectation of variable $*$. Considering that the KL-divergence function does not have an upper boundary, here we use the JS-divergence function to substitute the KL-divergence function in the first term [24]. Following the formulation in [91] and using the KL-divergence loss function proposed in Chapter 4 as the second term, the complete training objective of M2IOSR is converted from Eqn. (5.7) to a more clear and implementable formulation:

$$\begin{aligned}
& \min_{p(\mathbf{z}|\mathbf{x}), T(\mathbf{x}, \mathbf{z})} \left\{ -(1+\gamma) \left[E_{\mathbf{x} \sim p(\mathbf{z}|\mathbf{x})p(\mathbf{x})} (\log \sigma(T[\mathbf{x}, \mathbf{z}])) \right. \right. \\
& \quad \left. \left. + E_{\mathbf{x} \sim p(\mathbf{z})p(\mathbf{x})} (\log(1 - \sigma(T[\mathbf{x}, \mathbf{z}])) \right) \right] \\
& \quad \left. + \gamma \mathbb{E}_{\mathbf{x}_k \sim p(\mathbf{x}_k)} \left[D_{KL}(p(\mathbf{z}|\mathbf{x}_k)||q^{(k)}(\mathbf{z})) \right] \right\},
\end{aligned} \tag{5.8}$$

where σ is an activation function and T is a discriminator. The first term is the MI loss L_{MI} , which contains two parts: maximizing MI on positive pairs from the same input, and minimizing MI on negative pairs from different inputs. When $\mathbf{x} \sim p(\mathbf{z}|\mathbf{x})p(\mathbf{x})$, $[\mathbf{x}, \mathbf{z}]$ is a positive pair as the latent feature $\mathbf{z}$ is extracted from its corresponding input $\mathbf{x}$. While when $\mathbf{x} \sim p(\mathbf{z})p(\mathbf{x})$, $[\mathbf{x}, \mathbf{z}]$ is a negative pair as $\mathbf{z}$ is extracted from a random input. The second term is the KL-divergence loss L_{KL} . $p(\mathbf{z}|\mathbf{x}_k)$ is the k -th class conditional posterior distributions of latent features where k is the index of known classes, and $q^{(k)}(\mathbf{z})$ is the k -th multivariate Gaussian distribution $N(\mathbf{z}; \boldsymbol{\mu}_k, \mathbf{I})$.

5.3.2 Mutual Information Maximization

The overall architecture of the proposed method on positive pairs is shown in Figure 5.1. To have a fair comparison with previous works, the backbone used here is the re-designed VGG-13 network defined in [4], and the dimension of latent features is

fixed to 32. Here we use $f_d(\mathbf{x})$ to denote a feature with a spatial size of $d \times d$. Following the set-up in [24], the proposed method maximizes MI between the latent feature vector and the feature map after the first few layers (global MI) and maximizes average MI between the feature vector and every local region (patches) of the feature map (local MI), as well as encourages the representations to have desired characteristics.

FIGURE 5.1: Overview architecture of the proposed method on positive pairs. $f_d(\mathbf{x})$ represents a feature with a spatial size of $d \times d$. During the training phase, we maximize global MI on $[f_{16}(\mathbf{x}), f_1(\mathbf{x})]$ to extract useful information of the input, and maximize local MI on $[f_{16}(\mathbf{x}), f_1(\mathbf{x})]$, $[f_4(\mathbf{x}), f_1(\mathbf{x})]$ and $[f_4(\mathbf{x}), f_4(\mathbf{x})]$ to filter out useless features and prevent MI from being lost in the middle layers. Meanwhile, to further reduce open space risk [2], latent features $f_1(\mathbf{x})$ are constrained to class conditional Gaussian distributions. A 2-dimensional tSNE [1] visualization of $f_1(\mathbf{x})$ distributions on the CIFAR10 dataset [3] is shown at the bottom-left.

Therefore, the loss function used in M2IOSR is extended from Eqn. (5.4) to:

$$L = -(\beta_1 L_{global} + \beta_2 L_{local}) + \gamma L_{KL} \quad (5.9)$$

where β_1 , β_2 and γ are the balancing parameters

Global MI. Maximizing the MI between the whole input image and its corresponding latent feature helps the encoder extract useful information to pass through the network [24]. To this end, we encode the input x into the deep feature $f_{16}(x)$, whose spatial

size is 16×16 . $f_{16}(\mathbf{x})$ is then processed by two convolutional layers with the kernel size of 3×3 . The two layers' output is flattened as a vector to concatenate with the latent feature $f_1(\mathbf{x})$. The concatenated feature vector is input to a global discriminator T_g , which consists of two fully-connected layers, to get the global score. The activation function σ_g used here is a `Softplus` function. Details of the global discriminator architecture are listed in Table 5.1. For positive pairs, $f_{16}(\mathbf{x})$ and $f_1(\mathbf{x})$ are from the same image. Negative pairs are formed by linking the latent feature $f_1(\mathbf{x})$ with a $f_{16}(\hat{\mathbf{x}})$ from another image.

TABLE 5.1: Global discriminator architecture.

Layer	Size	Activation
Linear Layer	512	ReLU
Linear Layer	512	ReLU
Linear Layer	1	Softplus

Local MI. Maximizing the average MI between local patches of the input and the latent features helps the encoder filter out noisy features [24]. For this objective, $f_1(\mathbf{x})$ is concatenated with the feature map $f_{16}(\mathbf{x})$ at every location. We use two 1×1 convolutional layers as the local discriminator to score the local MI pair $[f_{16}(\mathbf{x}), f_1(\mathbf{x})]$ (l_{1t16}). The activation function σ_d used here is also a `Softplus` function. Details of the local discriminator architecture are listed in Table 5.2.

TABLE 5.2: Local discriminator architecture.

Layer	Dimension	Activation
1×1 Convolutional Layer	512	ReLU
1×1 Convolutional Layer	512	ReLU
1×1 Convolutional Layer	1	Softplus

Multi-scaled Local MI. To prevent MI from being lost in the middle layers, we form another two positive pairs $[f_4(\mathbf{x}), f_1(\mathbf{x})]$ (l_{1t4}), $[f_4(\mathbf{x}), f_4(\mathbf{x})]$ (l_{4t4}). Correspondingly, we also utilize three different negative pairs for Local MI during training: $[f_{16}(\hat{\mathbf{x}}), f_1(\mathbf{x})]$, $[f_4(\hat{\mathbf{x}}), f_1(\mathbf{x})]$, and $[f_4(\hat{\mathbf{x}}), f_4(\mathbf{x})]$, where $\hat{\mathbf{x}}$ and $\mathbf{x}$ are different images. The local MI loss L_{local} in Eqn. (5.9) is consisted of three terms:

$$L_{local} = a_1 l_{1t16} + a_2 l_{1t4} + a_3 l_{4t4} \quad (5.10)$$

where a_1 , a_2 and a_3 are the balancing parameters, and $a_1 + a_2 + a_3 = 1$.

Statistical Constraint. To further reduce the open space risk [2], we force latent features from different classes to approximate different multivariate Gaussian distributions $N(\mathbf{z}; \boldsymbol{\mu}_k, \mathbf{I})$ by using the following KL-divergence loss function proposed in Chapter 4:

$$\begin{aligned} & -D_{KL}(p(\mathbf{z}|\mathbf{x}, k) \parallel q^{(k)}(\mathbf{z})) \\ &= \frac{1}{2} \sum_{j=1}^J (1 + \log(\sigma_j^2) - (\mu_j - \mu_j^{(k)})^2 - \sigma_j^2) \end{aligned} \quad (5.11)$$

where $p(\mathbf{z}|\mathbf{x}, k)$ is the class conditional posterior distributions and $q^{(k)}(\mathbf{z})$ is k -th multivariate Gaussian model. A fully-connected layer is adopted to map the one-hot encoding vector of the input's label to the latent space, and the mean of k -th Gaussian distribution $\boldsymbol{\mu}_k$ is obtained. A two-dimensional tSNE [1] visualization of latent feature distributions of M2IOSR on the CIFAR10 dataset [3] is shown at the bottom-left of Figure 5.1. It is observed that latent features are clustered into 10 multivariate Gaussian distributions and each distribution represents one known class.

5.3.3 Training and Testing

Training. There are two phases during the M2IOSR training procedure: the *max-min* phase and the *classification* phase. During the *max-min* phase, following the loss function defined in Eqn. (5.9), the encoder and discriminators are trained jointly to maximize MI on positive pairs and minimize MI on negative pairs, as well as constrain the latent features to have desired statistical characteristics. During the *classification* phase, the encoder is trained to minimize the cross-entropy loss between predicted scores and ground-truth labels.

Testing. A given image is passed through the encoder network to obtain the probability score vector $\mathbf{V}$ on known classes. Following the set-up in [12], the given image will be identified as an unknown sample if its maximum predicted probability $\max(\mathbf{V})$ is smaller than a predetermined threshold τ .

TABLE 5.3: Comparison of testing accuracies on closed sets between vanilla CNN models and our proposed approach M2IOSR. The training objective of M2IOSR is to maximize mutual information as well as classify known classes, but there is no large decline in testing accuracies on closed sets.

Architecture	MNIST	SVHN	CIFAR10
Vanilla CNN	0.995	0.940	0.906
M2IOSR (ours)	0.995	0.935	0.903

5.4 Experiments

5.4.1 Implementation Details

To have a fair comparison with previous works, following [4], we employ the VGG-13 as our backbone, please refer to [4] for details. We set the dimension of latent features to 32. The proposed M2IOSR is trained end-to-end by an SGD optimizer with an initial learning rate of 0.01 (multiply by 0.1 for every 50 epochs) and a momentum of 0.9. Batch-size is set to 64. For the loss function defined in Eqn. (5.9), β_1 equals to 0.5, β_2 equals to 1.0, and γ equals to 0.1. For the local MI loss defined in Eqn. (5.10), a_1 equals to 0.7, a_2 equals to 0.1, and a_3 equals to 0.2. Networks trained by the proposed loss function should have comparable closed set testing accuracies with those trained only by the classification loss. The testing results of the closed set on each benchmark are shown in Table 5.3. The decision threshold τ is fixed to 0.95.

5.4.2 Dataset Description

We conduct OSR experiments on six popular benchmark datasets: MNIST [57], SVHN [92], CIFAR-10 [3], CIFAR-100 [83], ImageNet [84] and LSUN [85].

MNIST. MNIST [57] is the most popular hand-written digit benchmark. There are 60,000 images for training and 10,000 images for testing from ten digit classes. Each image is monochrome with a size of 28×28 . Although near 100% accuracy has been achieved in the closed set classification [86], the open set extension of MNIST still remains a challenge due to the variety of possible outliers.

SVHN. The Street View House Numbers (SVHN) [92] is a dataset consisting of ten digit classes each with between 9,981 and 11,379 color images with a size of 32×32 .

CIFAR-10. The CIFAR-10 dataset [3] consists of 5,000 training images and 1,000 testing images in each of the ten natural classes. Each image is colorful with a size of 32×32 . In CIFAR-10, each class has large intra-class diversities by color, style, or pose difference.

CIFAR-100. The CIFAR-100 dataset [83] is just like the CIFAR-10 dataset, except it has 100 natural classes containing 600 images each. There are 500 training images and 100 testing images in each class.

ImageNet. ImageNet [84] is a dataset of over 15 million labeled high-resolution images belonging to roughly 22,000 categories. The images were collected from the web and labeled by human labor.

LSUN. The LSUN dataset [85] contains around 10 million labeled images in 10 scene categories (around one million labeled images in each category) and 59 million labeled images in 20 object categories.

5.4.3 Ablation Analysis

In this section, we analyze the effectiveness of each module used in the proposed method on the CIFAR10 [3] and CIFAR100 dataset [83]. All classes in the CIFAR10 dataset are selected as known classes. Unknown classes are sampled from the CIFAR100 dataset varying from 10 to 100 (10, 14, 19, 25, 32, 42, 54, 71, 100), which means Openness level [2] varies from 18% to 59%. The performance is measured by the macro-average F1-scores [72] in all known classes and the unknown class against varying Openness levels.

We conduct experiments on the following baselines for ablation analysis. In all ablation experiments, the re-designed VGG-13 network defined in [4] is selected as the backbone, and if the maximum predicted score of input is less than the threshold τ defined in Section 5.4.1, this input will be classified as unknown.

I. CNN: A vanilla convolutional neural network (CNN) with a closed set classifier is trained for known classification. During testing, the classifier will give out the probability scores of known classes.

II. AE: An auto-encoder (AE) model is set up for known classification, with a closed set classifier on the top of learned latent features.

III. DIM: A deep infomax (DIM) [24] model is built up for ablation analysis, where the setting is nearly the same as the original DIM method: only the $f_{16}(\mathbf{x})$ feature map and $f_1(\mathbf{x})$ feature vector are used for mutual information maximization, and the latent features are constrained to have desired independence characteristics through adversarial learning. A closed set classifier is on the top of learned latent features, which is trained for known classification.

IV. DIM+KL: The model architecture and mutual information maximization procedure are the same as baseline III, but differently, the latent features are constrained to have conditional Gaussian distributions through the KL-divergence loss function proposed in Chapter 4 (Eqn. (5.11)).

V. DIM+KL+L1t4: The model architecture and statistical constraint procedure are the same as baseline IV, but l_{1t4} defined in Eqn. (5.10) is added to the local MI loss function. a_1 and a_2 are set equal to 0.7 and 0.3 respectively (as l_{4t4} is not used here, a_3 equals to 0).

VI. DIM+KL+L4t4: The model architecture and statistical constraint procedure are the same as baseline IV, but l_{4t4} defined in Eqn. (5.10) is added to the local MI loss function. a_1 and a_3 are set equal to 0.7 and 0.3 respectively (as l_{1t4} is not used here, a_2 equals to 0).

VII. Proposed Method: The model architecture and statistical constraint procedure are the same as baseline IV, but l_{1t4} and l_{4t4} are both added to the local MI loss function as Eqn. (5.10). a_1 , a_2 and a_3 are set equal to 0.7, 0.1 and 0.2 respectively. Besides, we also conduct ablative experiments when only using positive or negative pairs in the proposed method (namely **Positive** and **Negative** respectively in Figure 5.2).

The ablative results are drawn in Figure 5.2. The vanilla CNN model (baseline I) gives the worst performance, and the AE architecture (baseline II) only improves the performance a little, which shows the reconstruction strategy is indeed helpful for OSR but the effectiveness is limited. The overall performance in these two baselines is not so good as the F1-scores drop dramatically with Openness levels increasing. In baseline III, this trend is alleviated and the performance is increased by introducing MI maximization strategy, which shows the effectiveness of this strategy for OSR. In baseline IV, there is an obvious performance improvement when latent features are constrained to class conditional Gaussian distributions by the KL-divergence loss function defined in Eqn. (5.11). The overall performance is improved a little by maximizing local MI

FIGURE 5.2: F1-scores against varying Openness levels in ablative experiments. The proposed method outperforms other baselines and achieves the highest F1-scores on all Openness levels.

on the pair $[f_4(\mathbf{x}), f_1(\mathbf{x})]$ (baseline V) or $[f_4(\mathbf{x}), f_4(\mathbf{x})]$ (baseline VI). As a result, the proposed method (baseline VII) outperforms other baselines and achieves the highest F1-scores on all Openness levels. Noted that if we only use positive or negative pairs in the proposed method, the performance is comparable with the proposed method when the Openness level is small (less than 38%), but the performance degrades dramatically when the Openness rises to more than 38%.

5.4.4 Case Study

The class activation mapping (CAM) [93] is a technique used to observe what image regions are extracted features localized to. To compare localized regions between AE and the proposed method M2IOSR, we train an AE model and an M2IOSR model on the Dogs vs. Cats dataset on a VGG-19 network [6], where the last layer before latent features is replaced by a global pooling layer for CAM. The encoder architecture and parameter setting are the same in these two methods. Differently, in the AE model, we build up a decoder architecture and use a pixel-level reconstruction training strategy, while in the proposed method, we discard this decoder architecture and use the proposed loss function to train the network. Some class activation maps have been shown in the Chapter 1.2, and here we present more examples in Figure 5.3. Although the classification capabilities of these two methods are basically equivalent, the proposed method can extract more class-specific features and localize target objects more accurately, especially when the proportion of background is small. As the pixel-level reconstruction

strategy used in the AE training phase, features extracted from AE models also localize some stuff in the background, which is not conducive to OSR.

FIGURE 5.3: Class activation maps produced by auto-encoders and the proposed method M2IOSR.

We conduct another case-study on the CIFAR-10 dataset [3] where all animal classes (bird, cat, deer, dog, frog, and horse) are selected as known classes, and the unknown

FIGURE 5.4: Histograms of maximum prediction probability on the known testing set and unknown testing set for the AE model and M2IOSR. More unknown samples have smaller prediction probabilities in the M2IOSR, making it more effective to detect unknown samples through a threshold.

class set consists of four vehicle classes (airplane, automobile, ship, and truck). In this case study, we draw the histograms of maximum prediction probability on the known testing set and unknown testing set, as shown in Figure 5.4. For the AE model, most maximum prediction probabilities of unknown samples are close to 1 (basically equivalent to that of known samples), which makes it more difficult to distinguish between known and unknown samples. However, for the proposed method M2IOSR, more unknown samples have smaller prediction probabilities than that of known samples, making more unknown samples can be figured out through a threshold defined on the prediction scores. We also draw a tSNE [1] visualization on a 2-dimensional space for the proposed method M2IOSR, as shown in Figure 5.5. Known samples from different classes gather into different clusters, but most unknown samples are located in the open space.

Meanwhile, the confusion matrices produced by the AE model and the proposed method are shown in Figure 5.6. We divide each matrix into four parts by the red lines. It is observed that the results on known classes (the top-left part) are comparable, but the proposed method significantly increases the performance on detecting unknown samples (the bottom-right part) and reduces errors in classifying unknown samples into known categories (the bottom-left part). Figure 5.6 shows that our method can detect unknown samples more effectively than the AE model and maintain high accuracy in known classification.

FIGURE 5.5: tSNE [1] visualization of feature distributions of testing samples in M2IOSR. Samples in known classes are distributed into six clusters, while most unknown samples are located in the open space.

TABLE 5.4: Averaged F1-scores in five randomized trials.

Method	MNIST	SVHN	CIFAR10	CIFAR+10	CIFAR+50
Softmax	0.768	0.725	0.600	0.701	0.637
Openmax [16]	0.798	0.737	0.623	0.731	0.676
CROSR [4]	0.803	0.753	0.668	0.769	0.684
GDFR [12]	0.821	0.716	0.700	0.776	0.683
M2IOSR	0.892	0.788	0.733	0.796	0.744

5.4.5 Comparison with State-of-the-art Results

Here, we report our results on benchmarks and compare the proposed M2IOSR with existing OSR methods. An ideal OSR classifier should accurately classify known classes and effectively identify unknown samples, therefore in all experiments, the results are measured by macro-averaged F1-scores among all known classes and the unknown class.

The results of our M2IOSR are demonstrated in Table 5.4, 5.5 and 5.6. In Table 5.4, we conduct experiments on four standard image datasets: MNIST [57], SVHN [92], CIFAR10 [3] and CIFAR100 [83]. For each of the MNIST, SVHN, and CIFAR10 datasets, known classes consist of six randomly selected categories, and the remaining four categories constitute the unknown class. Meanwhile, in the CIFAR+10 /CIFAR+50 task, 4 vehicle categories in the CIFAR10 dataset (airplane, automobile, ship, and truck) are selected as known classes, and 10/50 animal categories in the CIFAR100 dataset are randomly selected to constitute the unknown class. The results shown in Table 5.4 are

		Prediction						
		Bird	Cat	Deer	Dog	Frog	Horse	Unknown
Ground-truth	Bird	808	12	13	8	10	1	148
	Cat	19	653	16	23	15	5	269
	Deer	10	5	823	1	2	6	153
	Dog	9	38	10	698	3	3	239
	Frog	7	7	3	2	885	0	96
	Horse	5	7	4	5	0	854	125
Ground-truth	Airplane	464	32	51	2	10	20	421
	Automobile	66	73	17	16	155	84	589
	Ship	205	79	85	3	53	20	555
	Truck	67	96	36	17	70	113	601

Auto-encoders

		Prediction						
		Bird	Cat	Deer	Dog	Frog	Horse	Unknown
Ground-truth	Bird	730	3	19	3	8	1	236
	Cat	16	545	16	32	14	4	373
	Deer	11	6	747	2	6	8	220
	Dog	5	34	8	646	3	3	301
	Frog	7	4	6	2	825	0	156
	Horse	4	5	3	2	0	796	190
Ground-truth	Airplane	390	23	37	2	7	21	520
	Automobile	73	43	15	13	67	69	720
	Ship	121	74	53	7	27	13	705
	Truck	37	68	13	18	19	133	712

Ours

FIGURE 5.6: **Confusion Matrix.** Here we show the confusion matrices produced by the AE model and our approach. The vehicle classes (airplane, automobile, ship, and truck) form the unknown class set and are not used in the training stage. Our approach performs much better on detecting unknown samples (the bottom-right part) than the AE model and significantly reduces errors in classifying unknown samples into known categories (the bottom-left part).

averaged F1-scores among 5 runs on different known and unknown separations. Noted that some previous works used Area under ROC curve (AUROC) to measure the performance on these tasks, here we used released codes to evaluate OSR performance on macro-averaged F1-scores and report these results in Table 5.4. As shown in this table, the proposed approach significantly outperforms existing OSR methods and achieves the highest averaged F1-scores.

In Table 5.5, following the protocol defined in [4], the CIFAR10 dataset consist of known classes, and two other datasets (ImageNet [84] and LSUN [85]) are respectively selected as the unknown class. To make the unknown sample size the same as known samples, we crop or resize unknown samples to 32×32 , then ImageNet-resize, ImageNet-crop, LSUN-resize, and LSUN-crop are generated. In each dataset, there are a total number of 10,000 testing images, which is the same as the CIFAR10 testing set. This makes the known-to-unknown ratio 1:1 during testing. The F1-scores of the proposed method and other OSR methods are shown in Table 5.5. We can see that the proposed method achieves new state-of-the-art performance. Although the proposed method improves the

TABLE 5.5: F1-scores on the CIFAR10 dataset with four other datasets added to the testing set as unknown.

Method	ImageNet crop	ImageNet resize	LSUN crop	LSUN resize
Softmax	0.639	0.653	0.642	0.647
Openmax [16]	0.660	0.684	0.657	0.668
LadderNet+Softmax [4]	0.640	0.646	0.644	0.647
LadderNet+Openmax [4]	0.653	0.670	0.652	0.659
DHRNet+Softmax [4]	0.645	0.649	0.650	0.649
DHRNet+Openmax [4]	0.655	0.675	0.656	0.664
CROSR [4]	0.721	0.735	0.720	0.749
C2AE [11]	0.837	0.826	0.783	0.801
GDFR [12]	0.757	0.792	0.751	0.805
M2IOSR	0.840	0.834	0.818	0.826

TABLE 5.6: F1-scores on the MNIST dataset with three other datasets added to the testing set as unknown.

Method	Omniglot	MNIST-noise	Noise
Softmax	0.595	0.801	0.829
Openmax [16]	0.780	0.816	0.826
CROSR [4]	0.793	0.827	0.826
M2IOSR	0.904	0.891	0.904

performance a little on the ImageNet dataset, the improvement on the LSUN dataset is significant.

In another experiment, following the set-up in [4], the complete MNIST training set consists of all known classes. Three other datasets, Omniglot [89], MNIST-Noise, and Noise are respectively selected as the unknown class. Omniglot is a dataset containing various alphabetic characters. Noise is a synthetic dataset, and each pixel value of its sample is independently from a uniform distribution between 0 and 1. MNIST noise is also a synthesized dataset by adding noise to the MNIST testing samples. Each unknown dataset has the same number of samples as the MNIST testing set, making the ratio of the known-to-unknown is 1:1. The F1-scores on these datasets are listed in Table 5.6. Our method achieves new state-of-the-art performance on all given tasks.

In Table 5.7, we measure the running time (milliseconds/image) of feature extraction and the number of parameters of each model when testing the CIFAR10 dataset on a single Tesla V100 graphic processor. Compared with other OSR methods, the proposed method

TABLE 5.7: Comparison of running time and number of parameters. All the models are based on VGG-13.

Method	Time	Parameter
Softmax	0.179	9,428k
Openmax [16]	0.181	9,428k
CROSR [4]	0.184	10,003k
GDFR [12]	0.772	10,424 k
M2IOSR	0.618	9,444 k

has a relatively small number of parameters. However, as we introduce positive/negative pairs and global/local MI maximization strategy, the highest F1-scores are achieved at a reasonable cost of longer running time.

5.5 Chapter Summary

In this chapter, we propose an OSR approach termed Maximal Mutual Information Open Set Recognition (M2IOSR). We analyze the pixel-level reconstruction strategy used in most previous OSR methods and find that this strategy is commonly over-demanding for OSR tasks and makes class-specific features contained in target objects less pronounced. To address these shortcomings, we discard the reconstruction strategy and propose an MI-based method with a streamlined architecture. The proposed M2IOSR only employs an encoder for class-specific feature extraction, thus is much more lightweight than previous methods with a decoder. Our multi-scaled MI maximization strategy encourages the encoder to establish strong interdependence between the input image and its latent features, which greatly improves the model’s ability to classify known classes and detect unknown samples. Moreover, to further reduce the open space risk, latent features are constrained to class conditional Gaussian distributions by a KL-divergence loss function. The proposed M2IOSR greatly enhances the performance of our baseline and leads to new state-of-the-art results on several standard benchmarks.

Chapter 6

Comparison of Proposed OSR Methods

In this chapter, we compare the proposed three OSR methods (*i.e.*, Discriminative Loss, CPGM and M2IOSR) on six standard image datasets (MNIST [57], SVHN [92], CIFAR-10 [3], CIFAR-100 [83], ImageNet [84], and LSUN [85]) and the air pressure data collected from a pneumatic train door subsystem.

6.1 Comparison on Image Datasets

In this section, we compare the proposed three OSR methods on six standard image datasets. To have a fair comparison among these methods, the backbone is fixed to the VGG-13 network defined in [4]. The batch size is fixed to 64 and the dimensionality of the latent feature is fixed to 32. The performance is measured by macro-averaged F_1 -scores [72]. We conduct experiments on six popular benchmark datasets: MNIST [57], SVHN [92], CIFAR-10 [3], CIFAR-100 [83], ImageNet [84] and LSUN [85]. These datasets are introduced as below:

MNIST. MNIST [57] is the most popular hand-written digit benchmark. There are 60,000 images for training and 10,000 images for testing from ten digit classes. Each image is monochrome with a size of 28×28 . Although near 100% accuracy has been achieved in the closed set classification [86], the open set extension of MNIST still remains a challenge due to the variety of possible outliers.

SVHN. The Street View House Numbers (SVHN) [92] is a dataset consisting of ten digit classes each with between 9,981 and 11,379 color images with a size of 32×32 .

TABLE 6.1: Comparison of closed set testing accuracies between the vanilla CNN and the proposed three methods.

Architecture	MNIST	SVHN	CIFAR-10
Plain CNN	0.995	0.940	0.906
Discriminative Loss	0.997	0.949	0.906
CPGM-VAE	0.996	0.942	0.912
CPGM-AAE	0.995	0.937	0.906
M2IOSR	0.995	0.935	0.903

CIFAR-10. The CIFAR-10 dataset [3] consists of 5,000 training images and 1,000 testing images in each of the ten natural classes. Each image is colorful with a size of 32×32 . In CIFAR-10, each class has large intra-class diversities by color, style, or pose difference.

CIFAR-100. The CIFAR-100 dataset [83] is just like the CIFAR-10 dataset, except it has 100 natural classes containing 600 images each. There are 500 training images and 100 testing images in each class.

ImageNet. ImageNet [84] is a dataset of over 15 million labeled high-resolution images belonging to roughly 22,000 categories. The images were collected from the web and labeled by human labor.

LSUN. The LSUN dataset [85] contains around 10 million labeled images in 10 scene categories (around one million labeled images in each category) and 59 million labeled images in 20 object categories.

The proposed methods should be trained without any large degradation in closed set accuracy from the original ones. The closed set accuracies of the proposed methods on MNIST, SVHN, and CIFAR-10 are listed in Table 6.1. We can see that although the proposed methods are designed for OSR, there is no large decline in testing accuracies on closed sets, and Discriminative Loss and CPGM-VAE even increase the closed set accuracies on these image datasets.

In Table 6.2, following the set-up in [4], the complete MNIST training set consists of all known classes. Three other datasets, Omniglot [89], MNIST-Noise, and Noise are respectively selected as the unknown class. Omniglot is a dataset containing various alphabetic characters. Noise is a synthetic dataset, and each pixel value of its sample is independently from a uniform distribution between 0 and 1. MNIST noise is also

TABLE 6.2: F_1 -scores on the MNIST dataset with three other datasets added to the testing set as unknown.

Method	Omniglot	MNIST-noise	Noise
Softmax	0.595	0.801	0.829
Openmax [16]	0.780	0.816	0.826
CROSR [4]	0.793	0.827	0.826
Discriminative Loss	0.791	0.816	0.821
CPGM-VAE	0.850	0.887	0.859
CPGM-AAE	0.872	0.865	0.872
M2IOSR	0.904	0.891	0.904

a synthesized dataset by adding noise to the MNIST testing samples. Each unknown dataset has the same number of samples as the MNIST testing set, making the ratio of the known-to-unknown is 1:1. The F_1 -scores on these datasets are listed in Table 6.2. It is shown that the proposed CPGM and M2IOSR methods achieve new state-of-the-art performance on these MNIST-based tasks.

In Table 6.3, following the protocol defined in [4], the CIFAR10 dataset consists of known classes, and two other datasets (ImageNet [84] and LSUN [85]) are respectively selected as the unknown class. To make the unknown sample size the same as known samples, we crop or resize unknown samples to 32×32 , then ImageNet-resize, ImageNet-crop, LSUN-resize, and LSUN-crop are generated. In each dataset, there are a total number of 10,000 testing images, which is the same as the CIFAR10 testing set. This makes the known-to-unknown ratio 1:1 during testing. The F_1 -scores of the proposed method and other OSR methods are shown in Table 6.3. We can see that the proposed CPGM and M2IOSR achieves new state-of-the-art performance on almost all given tasks (except CPGM-AAE on ‘ImageNet crop’ task). Although the proposed methods improve the performance a little on the ImageNet dataset, the improvement on the LSUN dataset is significant.

In Table 6.4, we conduct experiments on four standard image datasets: MNIST [57], SVHN [92], CIFAR-10 [3] and CIFAR-100 [83]. For each of the MNIST, SVHN, and CIFAR-10 datasets, known classes consist of six randomly selected categories, and the remaining four categories constitute the unknown class. Meanwhile, in the CIFAR+10 /CIFAR+50 task, 4 vehicle categories in the CIFAR-10 dataset (airplane, automobile, ship, and truck) are selected as known classes, and 10/50 animal categories in the CIFAR-100 dataset are randomly selected to constitute the unknown class. The

TABLE 6.3: F_1 -scores on the CIFAR-10 dataset with four other datasets added to the testing set as unknown.

Method	ImageNet crop	ImageNet resize	LSUN crop	LSUN resize
Softmax	0.639	0.653	0.642	0.647
Openmax [16]	0.660	0.684	0.657	0.668
CROSR [4]	0.721	0.735	0.720	0.749
C2AE [11]	0.837	0.826	0.783	0.801
GDFR [12]	0.757	0.792	0.751	0.805
Discriminative Loss	0.669	0.684	0.673	0.662
CPGM-VAE	0.840	0.832	0.806	0.812
CPGM-AAE	0.793	0.830	0.820	0.865
M2IOSR	0.840	0.834	0.818	0.826

TABLE 6.4: Averaged F_1 -scores in five randomized trials.

Method	MNIST	SVHN	CIFAR10	CIFAR+10	CIFAR+50
Softmax	0.768	0.725	0.600	0.701	0.637
Openmax [16]	0.798	0.737	0.623	0.731	0.676
CROSR [4]	0.803	0.753	0.668	0.769	0.684
GDFR [12]	0.821	0.716	0.700	0.776	0.683
Discriminative Loss	0.795	0.733	0.634	0.736	0.639
CPGM-VAE	0.837	0.776	0.655	0.760	0.695
CPGM-AAE	0.869	0.752	0.697	0.766	0.699
M2IOSR	0.892	0.788	0.733	0.796	0.744

results shown in Table 6.4 are averaged F_1 -scores among 5 runs on different known and unknown separations. As shown in this table, the proposed M2IOSR achieves the highest averaged F_1 -scores on all given tasks, and CPGM outperforms existing OSR methods on some tasks.

6.2 Comparison on Air Pressure Data

In this section, we compare the proposed OSR methods on the air pressure data collected from the pneumatic train door subsystem. The backbone used here is a VGG [6] inspired version, denoted as VGG-7, as shown in Figure 3.10. The network parameters are listed in Table 3.1. The batch size is fixed to 64 and the latent feature is fixed to 2. To evaluate the accuracy of OSR, here we use the F-measure values (Macro- F_1 and Micro- F_1) [72].

TABLE 6.5: Macro- F_1 and Micro- F_1 scores on the air pressure dataset for the extension phase and retraction phase.

Method	Extraction		Retraction	
	Macro- F_1	Micro- F_1	Macro- F_1	Micro- F_1
Eigenface[75]	0.631	0.563	0.553	0.526
Mahalanobis Distance[73]	0.897	0.854	0.808	0.782
Softmax[16]	0.711	0.499	0.680	0.460
Openmax[16]	0.857	0.831	0.781	0.625
Discriminative Loss	0.917	0.867	0.859	0.836
CPGM-VAE	0.969	0.977	0.889	0.867
CPGM-AAE	0.972	0.982	0.899	0.897
M2IOSR	0.900	0.847	0.852	0.818

The pressure data are collected at the interval of 0.1 seconds, during the two distinct phases of door operations: the extension phase (door opening) and the retraction phase (door closing). A typical pressure signal collected is shown in Figure 3.7. Based on operational experiences, four levels of air leakages: (*leakage 1, 2, 3, and 4*) were defined, in addition to normal operational data, i.e., no leakage. Figure 3.7 shows a typical air pressure signal collected by a pressure transducer. The data have been filtered and normalized to detect air leakages. We treat the door opening data, and door closing data separately, so that we have two similar classifiers, one for door opening and another for door closing. The plots of the five classes of data (filtered and normalized) are shown in Figure 3.8.

To feed collected air pressure data into the CNN, a 2D transformation is adopted and the input is transformed from time-series data to a 28x28 image (see Figure 3.9 and Eqn.(3.6)). A total of 51,200 samples are collected from *normal, leakage 1, 2, and 3* these four situations, and random 38,400 samples of them are chosen as the training set while the remaining 12,800 samples are chosen as the testing set. Meanwhile, 12,800 samples are collected under the *leakage 4* situation as unknown samples. They are then added to the testing set to constitute an open testing set.

The OSR results are shown in Table 6.5. All proposed methods achieve promising results, and CPGM-AAE achieves the highest Macro- F_1 and Micro- F_1 scores. It is worth noting that although M2IOSR achieves the state-of-the-art performance on standard image datasets, its F_1 scores on air pressure data are the lowest among proposed methods, which indicates that compared with Gaussian modeling (used in Discriminative Loss and CPGM), mutual information theory maybe not so applicable to such time-series data.

TABLE 6.6: Comparison of running time and number of parameters. All the models are based on VGG-13.

Method	Time	Parameter
Softmax	0.179	9,428k
Openmax [16]	0.181	9,428k
CROSR [4]	0.184	10,003k
GDFR [12]	0.772	10,424 k
Discriminative Loss	0.181	9,428k
CPGM-VAE	0.996	291,352k
CPGM-AAE	0.657	13,365k
M2IOSR	0.618	9,444 k

6.3 Comparison of Running Time

In practical applications, besides accuracy, running time is another important indicator to evaluate system performance. In Table 6.6, we measure the running time (millisecond/image) of feature extraction and the number of parameters of each model when testing the CIFAR10 dataset on a single Tesla V100 graphic processor. GPU is used both in the training phase and the testing phase. The memory size is 128 GB. The speed of the CPU is 2.5 GHz. Compared with other OSR methods, the proposed Discriminative Loss has a minimal number of parameters, and its running time is only 0.2 milliseconds per input slower than that of the fastest method Softmax. It is also worth noting that the M2IOSR has a relatively small number of parameters, and achieves the highest F_1 -scores on image datasets at a reasonable cost of longer running time.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we have explored how to enhance Open Set Recognition (OSR) performance from different aspects. Firstly we introduce three challenges in OSR, and proposed three OSR methods to address these challenges.

Firstly, although many OSR methods achieve promising results in many computer vision tasks, most of them have a long running time and are not applicable to real-time industrial tasks. To have a shorter running time, we proposed an OSR method, called Discriminative Loss. We combine the proposed loss function with the Softmax loss function, which is used in most Convolutional Neural Networks (CNNs), to force learned features in different classes to be close to different centroids for Gaussian modeling. The proposed method is demonstrated on the MNIST dataset and applied to detect different levels of air leakage on the pneumatic train door subsystem. Moreover, the proposed method has a faster running time compared with other deep learning-based OSR methods. More experiments have also been conducted to verify the stability of the proposed method across a wide range of the balancing parameter λ . The developed system provides a feasible and reliable solution for air leakage detection, and it seems to be the potential for the detection of other faults that may appear on the pneumatic train door subsystem.

Then, to increase OSR accuracy on standard image datasets, we propose a novel OSR method, called Conditional Probabilistic Generative Models (CPGM). The core insight

of this proposed method is to add discriminative information into probabilistic generative models (*i.e.*, Variational Auto-encoders (VAE) and Adversarial Auto-encoders (AAE)), such that the proposed models cannot only detect unknown samples but also classify known classes by forcing different latent features to approximate conditional Gaussian distributions. We discuss many model variants and provide comprehensive experiments to study their characteristics. Experiments on several standard image datasets show that the proposed methods significantly outperform the baseline methods. A case study between CPGM-VAE and CPGM-AAE is also conducted to investigate and compare their characteristics for OSR: CPGM-AAE could generate more compact known clusters for OSR and have a more clear decision boundary between known and unknown classes.

From above discussion, Auto-encoders (AE) is a popular model to generate latent features with desired statistical characteristics. Meanwhile, AE models also can extract class-specific features through their reconstruction training strategy. This reconstruction strategy requires the network to restore the input image on pixel-level. This strategy is commonly over-demanding for OSR since class-specific features are generally contained in target objects, not in all pixels. To address this shortcoming, we propose a mutual information (MI)-based method with a streamlined architecture, Maximal Mutual Information Open Set Recognition (M2IOSR). The proposed M2IOSR only uses an encoder to extract class-specific features by maximizing the MI between the given input and its latent features across multiple scales. Meanwhile, to further reduce the open space risk, latent features are constrained to class conditional Gaussian distributions by a KL-divergence loss function. The proposed M2IOSR greatly enhances the performance of our baseline and leads to new state-of-the-art results on several standard benchmarks.

Finally, we compare the proposed three methods on six standard image datasets and the air pressure data collected from a pneumatic train door subsystem. Although these proposed methods are designed for OSR, there is no large decline in testing accuracies on closed sets, and Discriminative Loss and CPGM-VAE even increase the closed set accuracies (see Table 6.1). For experiments on standard image datasets, CPGM significantly outperforms the baseline methods, and M2IOSR achieves the state-of-the-art performance (see Table 6.2, Table 6.3, and Table 6.4). For experiments on air pressure data, Gaussian-based models (Discriminative Loss, CPGM-VAE, and CPGM-AAE) all achieve promising results. Among them, CPGM-AAE achieves the highest F_1 scores in both the extension phase and the retraction phase (see Table 6.5). However, M2IOSR does not perform so well on the air pressure dataset, and its F_1 scores are even lower

than that of Discriminative Loss, which indicates that compared with Gaussian modeling, mutual information theory maybe not so applicable to such time-series data. It is also worth noting that compared with other OSR methods, Discriminative Loss has a minimal number of parameters, and its running time is only 0.2 milliseconds per input slower than that of the fastest method Softmax. M2IOSR has a relatively small number of parameters, and achieves the highest F1-scores on image datasets at a reasonable cost of longer running time (see Table 6.6).

7.2 Future Work

The proposed methods and the related theories have shown their advantages and potential in conducting OSR on image data in the computer vision field and time-series data in the industrial field. The future works that are worth exploring are summarized as follows:

- **Open World Recognition.** Unsupervised learning can be applied to those detected unknown samples, then these samples will be labeled with new classes. These new labeled samples will be incrementally incorporated into the classifier by incremental learning methods. Therefore, this classifier can be continuously updated and strengthened with additional object categories in the real world [31].
- **OSR in other tasks.** Current OSR methods mostly focus on image classification tasks [4] [11] [12] [35] [34]. In this thesis, we also apply OSR methods to fault detection on time-series data in the industrial field. In these years, many researchers study on developing OSR technique in other tasks, such as semantic segmentation [94], few shot learning [95], long-tailed recognition [96], and 3-dimensional cloud point classification [97]. We will explore to develop and apply OSR methods to other tasks in the future.
- **Real-time OSR.** Current OSR methods have achieved promising accuracies on standard image datasets, but consume too much resources, such as memory and running time. It is unrealistic to apply these OSR methods in real-world applications. To achieve effective and efficient OSR methods, the speed and the number of parameters in the networks must be taken into account. In this thesis, we propose Discriminative Loss, which is an OSR method with a short running time, but

its performance on image datasets is not satisfying. To realize real-time OSR on image datasets, we will explore shortening running time without a large decline in accuracy.

Author's Publications

Published or Accepted:

- **X. Sun**, K. V. Ling, K. K. Sin and Y. Liu. *Open Set Recognition Method for Air Leakage Detection of Pneumatic Train Door Subsystem*. IEEE Transactions on Instrumentation and Measurement, Paper No. TIM-21-00333.(Accepted)
- **X. Sun**, Z. Yang, C. Zhang, K. V. Ling and G. Peng. *Conditional gaussian distribution learning for open set recognition*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020 (pp. 13480-13489).(Published)
- **X. Sun**, K. V. Ling, K. K. Sin and L. Tay. *Intelligent Fault Detection and Diagnosis of Air Leakage on Train Door*. In 2018 International Conference on Intelligent Rail Transportation (pp. 1-4). IEEE.(Published)
- **X. Sun**, K. V. Ling, K. K. Sin and L. Tay. *Extreme Learning Machine Based Intelligent Condition Monitoring System on Train Door*. In International Conference on Extreme Learning Machine (pp. 98-107). Springer, Cham.(Published)
- Y. Liu, **X. Sun** and J. H. L. Pang. *A YOLOv3-based Deep Learning Application Research for Condition Monitoring of Rail Thermite Welded Joints*. In Proceedings of the 2020 2nd International Conference on Image, Video and Signal Processing (pp. 33-38).(Published)

Under Review:

- **X. Sun**, H. Ding, C. Zhang, G. Lin and K. V. Lin. *M2IOSR: Maximum Mutual Information Open Set Recognition*. IEEE Transactions on Image Processing (TIP-25736-2021).(Under Review)

Bibliography

- [1] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. [xxi](#), [xxii](#), [59](#), [60](#), [69](#), [71](#), [77](#), [78](#)
- [2] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772, 2012. [xxii](#), [1](#), [2](#), [6](#), [13](#), [14](#), [17](#), [18](#), [34](#), [52](#), [55](#), [64](#), [67](#), [69](#), [71](#), [73](#)
- [3] Alex Krizhevsky and Geoff Hinton. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7):1–9, 2010. [xxii](#), [51](#), [69](#), [71](#), [72](#), [73](#), [76](#), [78](#), [83](#), [84](#), [85](#)
- [4] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4016–4025, 2019. [xxiii](#), [2](#), [3](#), [6](#), [14](#), [17](#), [18](#), [19](#), [35](#), [50](#), [57](#), [58](#), [64](#), [68](#), [72](#), [73](#), [78](#), [79](#), [80](#), [81](#), [83](#), [84](#), [85](#), [86](#), [88](#), [91](#)
- [5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016. [1](#)
- [6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. [1](#), [9](#), [28](#), [75](#), [86](#)
- [7] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 4700–4708, 2017. [1](#)
- [8] Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126, 2004. [1](#), [18](#), [20](#)
- [9] Sergey Demyanov Zongyuan Ge and Rahil Garnavi. Generative openmax for multi-class open set classification. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 42.1–42.12. BMVA Press, September 2017. [2](#), [3](#), [6](#), [14](#), [17](#), [18](#), [19](#), [64](#)

- [10] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 613–628, 2018. [2](#), [3](#), [6](#), [14](#), [17](#), [18](#), [19](#), [64](#)
- [11] Poojan Oza and Vishal M Patel. C2ae: Class conditioned auto-encoder for open-set recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2307–2316, 2019. [2](#), [3](#), [6](#), [14](#), [17](#), [18](#), [19](#), [46](#), [58](#), [64](#), [80](#), [86](#), [91](#)
- [12] Pramuditha Perera, Vlad I Morariu, Rajiv Jain, Varun Manjunatha, Curtis Wiggington, Vicente Ordonez, and Vishal M Patel. Generative-discriminative feature representations for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11814–11823, 2020. [2](#), [3](#), [6](#), [14](#), [17](#), [35](#), [63](#), [64](#), [67](#), [71](#), [78](#), [80](#), [81](#), [86](#), [88](#), [91](#)
- [13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. [2](#), [17](#), [19](#), [38](#)
- [14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. [3](#), [10](#), [12](#), [17](#), [19](#), [37](#), [39](#), [40](#)
- [15] Lei Shu, Hu Xu, and Bing Liu. Doc: Deep open classification of text documents. In *Proceedings of Empirical Methods in Natural Language Processing*, 2017. [3](#), [14](#), [17](#), [18](#), [19](#)
- [16] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016. [3](#), [14](#), [17](#), [18](#), [19](#), [32](#), [33](#), [34](#), [35](#), [58](#), [78](#), [80](#), [81](#), [85](#), [86](#), [87](#), [88](#)
- [17] Caglar Aytekin, Xingyang Ni, Francesco Cricri, and Emre Aksu. Clustering and unsupervised anomaly detection with l2 normalized deep auto-encoder representations. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2018. [3](#), [14](#), [37](#), [38](#)
- [18] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International Conference on Machine Learning (ICML)*, pages 4393–4402, 2018. [3](#), [14](#), [37](#), [38](#)
- [19] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674, 2017. [3](#), [14](#), [37](#)
- [20] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018. [3](#), [14](#), [37](#), [38](#)

- [21] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. In *Advances in neural information processing systems*, pages 6822–6833, 2018. [3](#), [15](#), [37](#)
- [22] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. [3](#), [12](#), [37](#), [38](#), [39](#), [41](#)
- [23] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIG-MOBILE mobile computing and communications review*, 5(1):3–55, 2001. [6](#), [63](#), [64](#), [65](#)
- [24] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. [6](#), [16](#), [64](#), [65](#), [67](#), [68](#), [69](#), [70](#), [74](#)
- [25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. [9](#)
- [26] Walter J Scheirer, Lalit P Jain, and Terrance E Boulton. Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2317–2324, 2014. [13](#), [17](#)
- [27] Lalit P Jain, Walter J Scheirer, and Terrance E Boulton. Multi-class open set recognition using probability of inclusion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 393–409. Springer, 2014. [13](#), [17](#)
- [28] Hakan Cevikalp and Hasan Serhan Yavuz. Fast and accurate face recognition with image sets. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1564–1572, 2017. [13](#)
- [29] Hakan Cevikalp and Bill Triggs. Polyhedral conic classifiers for visual object detection and classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 261–269, 2017. [13](#)
- [30] He Zhang and Vishal M Patel. Sparse representation-based open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1690–1696, 2016. [13](#), [17](#)
- [31] Abhijit Bendale and Terrance Boulton. Towards open world recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1893–1902, 2015. [13](#), [17](#), [91](#)
- [32] Pedro R Mendes Júnior, Roberto M de Souza, Rafael de O Werneck, Bernardo V Stein, Daniel V Pazinato, Waldir R de Almeida, Otávio AB Penatti, Ricardo da S Torres, and Anderson Rocha. Nearest neighbors distance ratio open-set classifier. *Machine Learning*, 106(3):359–386, 2017. [13](#), [17](#)

- [33] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. [14](#)
- [34] Hongjie Zhang, Ang Li, Jie Guo, and Yanwen Guo. Hybrid models for open set recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. [14](#), [17](#), [91](#)
- [35] Guangyao Chen, Limeng Qiao, Yemin Shi, Peixi Peng, Jia Li, Tiejun Huang, Shiliang Pu, and Yonghong Tian. Learning open set network with discriminative reciprocal points. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. [14](#), [17](#), [91](#)
- [36] David MJ Tax and Robert PW Duin. Support vector domain description. *Pattern recognition letters*, 20(11-13):1191–1199, 1999. [14](#)
- [37] Larry M Manevitz and Malik Yousef. One-class svms for document classification. *Journal of machine Learning research*, 2(Dec):139–154, 2001. [14](#)
- [38] Stephen Roberts and Lionel Tarassenko. A probabilistic resource allocating network for novelty detection. *Neural Computation*, 6(2):270–284, 1994. [14](#)
- [39] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017. [15](#)
- [40] Yaoyao Liu, An-An Liu, Yuting Su, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. *arXiv preprint arXiv:2002.10211*, 2020. [15](#)
- [41] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover’s distance and structured classifiers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. [15](#)
- [42] Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9587–9595, 2019. [15](#)
- [43] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5217–5226, 2019. [15](#)
- [44] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. Crnet: Cross-reference networks for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4165–4173, 2020. [15](#)

- [45] Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988. [16](#)
- [46] Suzanna Becker. *An information-theoretic unsupervised learning algorithm for neural networks*. University of Toronto, 1992. [16](#)
- [47] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995. [16](#)
- [48] Suzanna Becker. Mutual information maximization: models of cortical self-organization. *Network: Computation in neural systems*, 7(1):7–31, 1996. [16](#)
- [49] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of International Conference on Machine Learning (ICML)*, 2017. [16](#)
- [50] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. [16](#)
- [51] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9865–9874, 2019. [16](#)
- [52] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. In *Proceedings of International Conference on Machine Learning (ICML)*, 2018. [16](#)
- [53] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems (NIPS)*, pages 15535–15545, 2019. [16](#), [65](#)
- [54] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. [16](#)
- [55] David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884, 2020. [16](#)
- [56] Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A Alemi, and George Tucker. On variational bounds of mutual information. In *Proceedings of International Conference on Machine Learning (ICML)*, 2019. [16](#)
- [57] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, 2:18, 2010. [17](#), [20](#), [21](#), [51](#), [72](#), [78](#), [83](#), [85](#)

- [58] Rui Zhao, Ruqiang Yan, Zhenghua Chen, Kezhi Mao, Peng Wang, and Robert X Gao. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115:213–237, 2019. [18](#)
- [59] Mingming Zhang, Chao Chen, Tianyu Wo, Tao Xie, Md Zakirul Alam Bhuiyan, and Xuelian Lin. Safedrive: online driving anomaly detection from large-scale vehicle data. *IEEE Transactions on Industrial Informatics*, 13(4):2087–2096, 2017. [18](#)
- [60] Dumidu Wijayasekara, Ondrej Linda, Milos Manic, and Craig Rieger. Mining building energy management system data using fuzzy anomaly detection and linguistic descriptions. *IEEE Transactions on Industrial Informatics*, 10(3):1829–1840, 2014. [18](#)
- [61] Moon-Hwan Chang, Chaochao Chen, Diganta Das, and Michael Pecht. Anomaly detection of light-emitting diodes using the similarity-based metric test. *IEEE Transactions on industrial informatics*, 10(3):1852–1863, 2014. [18](#)
- [62] Mingrui Fu, Jinhai Liu, Dong Zang, and Senxiang Lu. Anomaly detection of complex mfl measurements using low-rank recovery in pipeline transportation inspection. *IEEE Transactions on Instrumentation and Measurement*, 69(9):6776–6786, 2020. [18](#)
- [63] Chunjie Zhou, Shuang Huang, Naixue Xiong, Shuang-Hua Yang, Huiyun Li, Yuanqing Qin, and Xuan Li. Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(10):1345–1360, 2015. [18](#)
- [64] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29:2234–2242, 2016. [19](#)
- [65] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998. [20](#), [21](#)
- [66] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049, 2017. [20](#), [21](#)
- [67] David MJ Tax and Robert PW Duin. Support vector domain description. *Pattern recognition letters*, 20(11-13):1191–1199, 1999. [20](#)
- [68] F Turgis, R Copin, P Loslever, L Cauffriez, and N Caouder. Design of a testing bench for simulating tightened-up operating conditions of train’s passenger access. In *European Safety and Reliability Conference (ESREL)*, pages 21–23, 2009. [23](#)
- [69] Laurent Cauffriez, Pierre Loslever, Nathalie Caouder, Fabien Turgis, and Reynald Copin. Robustness study and reliability growth based on exploratory design of experiments and statistical analysis: a case study using a train door test bench.

- The International Journal of Advanced Manufacturing Technology*, 66(1-4):27–44, 2013. [23](#)
- [70] Tushhar Bora, Krushna Rajput, Swapnil Vikhe, Inamulhaq Shaikh, and Sahil Shinde. A review on pneumatic operated train door system. *International Research Journal of Engineering and Technology*, 6(01):472–477, 2019. [23](#)
- [71] S Preston. The 68–95–99.7 rule for normal distributions. *World Wide Web Electronic Publication*, 142, 2010. [29](#)
- [72] Yutaka Sasaki et al. The truth of the f-measure. *Teach Tutor mater*, 1(5):1–5, 2007. [31](#), [52](#), [55](#), [73](#), [83](#), [86](#)
- [73] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. National Institute of Science of India, 1936. [33](#), [34](#), [35](#), [87](#)
- [74] Jason Weston and Chris Watkins. Multi-class support vector machines. Technical report, Citeseer, 1998. [33](#)
- [75] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991. [33](#), [34](#), [35](#), [87](#)
- [76] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016. [33](#)
- [77] Weiyang Xie, Jie Lei, Baozhu Liu, Yunsong Li, and Xiuping Jia. Spectral constraint adversarial autoencoders approach to feature representation in hyperspectral anomaly detection. *Neural Networks*, 119:222–234, 2019. [38](#)
- [78] Jingyao Wu, Zhibin Zhao, Chuang Sun, Ruqiang Yan, and Xuefeng Chen. Fault-attention generative probabilistic adversarial autoencoder for machine anomaly detection. *IEEE Transactions on Industrial Informatics*, 2020. [38](#)
- [79] Samir Puuska, Tero Kokkonen, Janne Alatalo, and Eppu Heilimo. Anomaly-based network intrusion detection using wavelets and adversarial autoencoders. In *International Conference on Security for Information Technology and Communications*, pages 234–246. Springer, 2018. [38](#)
- [80] Nanjun Li and Faliang Chang. Video anomaly detection and localization via multi-variate gaussian fully convolution adversarial autoencoder. *Neurocomputing*, 369: 92–105, 2019. [38](#)
- [81] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1–9, 2015. [39](#)

- [82] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. How to train deep variational autoencoders and probabilistic ladder networks. In *33rd International Conference on Machine Learning (ICML)*, 2016. [39](#), [43](#), [45](#), [50](#)
- [83] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. [51](#), [52](#), [72](#), [73](#), [78](#), [83](#), [84](#), [85](#)
- [84] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. [51](#), [58](#), [72](#), [73](#), [79](#), [83](#), [84](#), [85](#)
- [85] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. [51](#), [52](#), [58](#), [72](#), [73](#), [79](#), [83](#), [84](#), [85](#)
- [86] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12):3207–3220, 2010. [51](#), [72](#), [83](#)
- [87] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. [54](#)
- [88] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017. [55](#)
- [89] Simon Ager. Omniglot - writing systems and languages of the world. 1998. URL <http://www.omniglot.com/>. [57](#), [80](#), [84](#)
- [90] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006. [63](#)
- [91] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems (NIPS)*, pages 271–279, 2016. [68](#)
- [92] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. [72](#), [78](#), [83](#), [85](#)
- [93] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2921–2929, 2016. [75](#)
- [94] Zhiying Cui, Wu Longshi, and Ruixuan Wang. Open set semantic segmentation with statistical test and adaptive threshold. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020. [91](#)

-
- [95] Bo Liu, Hao Kang, Haoxiang Li, Gang Hua, and Nuno Vasconcelos. Few-shot open-set recognition using meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2020. [91](#)
 - [96] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. [91](#)
 - [97] Ayush Bhardwaj, Sakshee Pimpale, Saurabh Kumar, and Biplab Banerjee. Empowering knowledge distillation via open set recognition for robust 3d point cloud classification. *arXiv preprint arXiv:2010.13114*, 2020. [91](#)