Deep Contrastive Metric Learning to Detect Polymicrogyria in Pediatric Brain MRI

by

Lingfeng Zhang

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the degree of
Master of Computer Science
Concentration in Applied Artificial Intelligence

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Declaration

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of University of Ottawa's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to IEEE website to learn how to obtain a License from RightsLink.

Abstract

Polymicrogyria (PMG) is one brain disease that mainly occurs in the pediatric brain. Heavy PMG will cause seizures, delayed development, and a series of problems. For this reason, it is critical to effectively identify PMG and start early treatment. Radiologists typically identify PMG through magnetic resonance imaging scans. In this study, we create and open a pediatric MRI dataset (named PPMR dataset) including PMG and controls from the Children's Hospital of Eastern Ontario (CHEO), Ottawa, Canada. The difference between PMG MRIs and control MRIs is subtle and the true distribution of the features of the disease is unknown. Hence, we propose a novel center-based deep contrastive metric learning loss function (named cDCM Loss) to deal with this difficult problem. Cross-entropy-based loss functions do not lead to models with good generalization on small and imbalanced dataset with partially known distributions. We conduct exhaustive experiments on a modified CIFAR-10 dataset to demonstrate the efficacy of our proposed loss function compared to cross-entropy-based loss functions and the state-of-the-art Deep SAD loss function. Additionally, based on our proposed loss function, we customize a deep learning model structure that integrates dilated convolution, squeeze-and-excitation blocks and feature fusion for our PPMR dataset, to achieve 92.01% recall. Since our suggested method is a computer-aided tool to assist radiologists in selecting potential PMG MRIs, 55.04% precision is acceptable. To our best knowledge, this research is the first to apply machine learning techniques to identify PMG only from MRI and our innovative method achieves better results than baseline methods.

Our code will be available at: https://github.com/RichardChangCA/Deep-Contrastive-

Metric-Learning-Method-to-Detect-Polymicrogyria-in-Pediatric-Brain-MRI.

Our pediatric MRI dataset will be available at: https://www.kaggle.com/datasets/lingfengzhang/pediatric-polymicrogyria-mri-dataset.

Acknowledgements

First of all, I am much obliged to my thesis supervisor, Professor Jochen Lang. I would appreciate his continuous support, encouragement, guidance, suggestions, and inspiration during my Master's study and research. Dr. Lang helps me avoid wrong research directions and lead to the correct research direction. He is enthusiastic to assist my research progress, so I can finish my Master's thesis successfully.

I would like to express my appreciation to Dr. Nishard Abdeen from Children's Hospital of Eastern Ontario(CHEO), Ottawa, Canada. He and the hospital provide data for us to carry out this research. Dr. Abdeen helps us annotate the dataset, which is a tremendous amount of work. Thanks for his patience and hard work. Moreover, he provides us with a medical background and gives us crucial suggestions throughout the whole research.

Moreover, I would like to thank my colleagues in the VIVA lab, my classmates and friends at UOttawa, and others who have a wonderful discussion and brainstorming with me to inspire my thesis research.

Dedication

First of all, I would like to thank my parents to give me tremendous financial and mental support during my whole Master's study. Without them, I cannot achieve anything in Canada.

I would also like to thank my friends, professors, classmates, mentors, and colleagues who help me a lot mentally and academically during my study in Canada.

Table of Contents

List of Tables

List of Figures

List of Acronyms

AE	Autoencoder
ANDI	Alzheimer's Disease National Initiative
ANN	Artificial Neural Networks
AUC	area under the curve
AUCROC	area under the curve of receiver operating characteristic
BCE	binary cross-entropy loss
cDCM	center-based deep contrastive metric loss
CE	cross entropy
CHEO	Children's Hospital of Eastern Ontario
CLBP	completed local binary patterns
CNN	convolutional neural network
CT	computed tomography
CV	cross validation
DCAE	deep convolutional autoencoder
DCGAN	deep convolutional generative adversarial networks

Deep SAD	deep semi-supervised anomaly detection
DWT	discrete wavelet transform
GAN	Generative Adversarial Networks
GAP	global average pooling
GLCM	gray level co-occurrence matrix
GPU	graphics processing unit
KNN	K-nearest neighbour
KPCA	kernel principle component analysis
LBP	local binary patterns
LDA	discriminate analysis
MLP	multi-layer perceptron
MRI	magnetic resonance imaging
MSE	mean square error
NAN	not a number
NAS	neural architecture search
OCSVM	one-class support vector machine
OCTID	one-class tumor image detection
OOD	out of distribution
PACS	picture archiving and communication system
PCA	principal component analysis
PMG	polymicrogyria
PPMR	pediatric polymicrogyria magnetic resonance image
ReLU	rectified linear unit

RF	Radio Frequency
ResNet	Residual neural network
ROC	receiver operating characteristic
ROI	region of interest
SE	squeeze and excitation
SVD	singular value decomposition
SVDD	support vector data description
SVM	support vector machine
TE	Time to Echo
TR	Repetition Time
UMAP	uniform manifold approximation and projection
VAE	Variational Autoencoder
WBCE	weighted binary cross-entropy loss

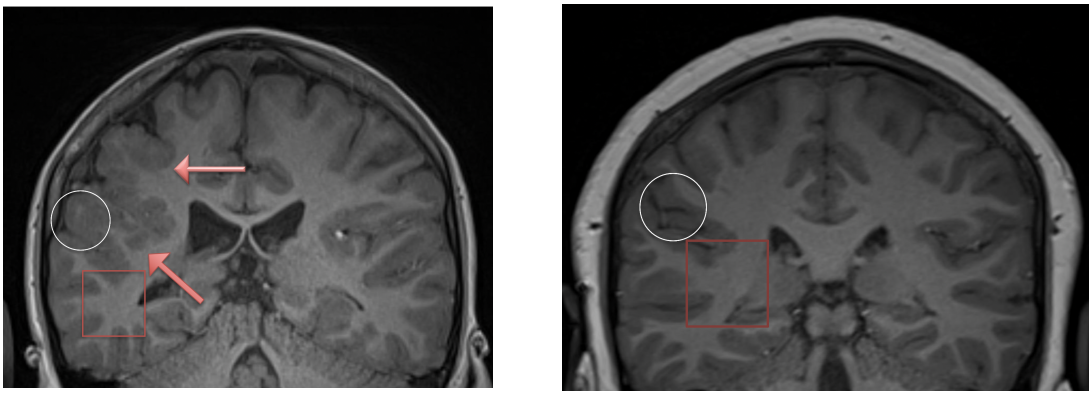
Chapter 1

Introduction

1.1 Problem Statement

Polymicrogyria (PMG) is a cortical malformation characterized by irregular and thickened grey matter, numerous small gyri, shallow sulci, and the decrease of grey-white matter differentiation. One example is shown in Fig 1.1. In Fig 1.1(a), PMG (arrows) is defined as thickened and irregular grey matter (circle), which has a more ill-defined and irregular interface with the brighter white matter (rectangle). In contrast, in Fig 1.1(b), a normal brain has relatively uniform smooth grey matter (circle) which has a well-defined border with white matter (rectangle). Although it can be caused by a congenital viral infection [7], PMG is one of the most frequent anomalies of cortical development that is often associated with genetic diseases. A heterogeneous condition, PMG can vary in severity, extent, and distribution [7]. Since PMG is a frequent cause of severe or treatment-resistant

epilepsy, the affected area might need to be surgically removed [8]. It can also be linked to serious symptoms like developmental delay and weakness on one side of the body or in the limbs [7]. Magnetic resonance imaging (MRI) is a reliable tool which can be used by skilled neuroradiologists to diagnose PMG, although it can miss focal or less obvious PMG. However, because PMG can be subtle, general radiologists might not be able to detect it with the same accuracy. Accurately diagnosing PMG is crucial since it can result from a variety of single gene mutations and disorders, and its existence would suggest genome sequencing and counselling. Therefore, a computer-aided tool should be developed to assist radiologists in identifying PMG from MRIs.



(a) PMG MRI (b) Control MRI

Figure 1.1: Difference between PMG and control MRI. Red arrows point to important features of PMG. The white circle on the left image shows thickened and irregular grey matter with numerous small gyri and shallow sulci, while the white circle on the right image shows regular grey matter. The red rectangle on the left image shows an ill-defined and irregular interface between white and grey matter, while the red rectangle on the right image shows a well-defined border.

1.2 Thesis Statement

Cortical malformations in the pediatric brain have been studied using machine learning, primarily in the context of fetal imaging [9] and epilepsy [10]. Machine learning might help with computer-aided diagnosis, enhancing the diagnostic accuracy of MRI. However, there is a dearth of research on the detection of PMG using machine learning, particularly deep learning. As opposed to brain tumour classification tasks [11], which have rather distinct tumour boundaries and actual ground truth from brain tumour segmentation tasks has been established, the diffuse and variable nature of PMG, as well as vague boundaries, are potential obstacles.

It is difficult to collect sufficient PMG MRIs in the real-world clinical scenario, although collecting normal pediatric brain MRI is relatively easy. Dr. Nishard Abdeen collected and annotated a pediatric PMG MRI dataset from Children's Hospital of Eastern Ontario (CHEO), Ottawa, Canada. This dataset is small because it only includes MRI from 23 different patients. In addition, this dataset is also imbalanced because of the majority of normal MRI. This imbalance originates from two sources: On the one hand control MRI are easier to obtain but on the other hand even for PMG samples the majority of MRI slices do not show evidence of PMG. In this thesis, we introduce this dataset in detail. Since this dataset is small and lacks the diversity of PMG, it is difficult to evaluate testing results. Therefore, we also introduce a modified CIFAR-10 dataset which mimics the main challenges of our PPMR dataset, including the fact that the minority class samples in the training data are not representative and that the whole dataset is imbalanced.

Conventional machine learning methods with traditional image processing based fea-

ture extraction methods appear not to work because the differences between normal and PMG MRI are subtle. Deep learning models with cross-entropy-based loss functions also perform poorly on this small imbalanced dataset due to their insufficient ability to generalize on unseen features. In this thesis, we investigate the generalization ability of deep learning models trained with cross-entropy-based loss functions on the modified CIFAR-10 dataset, mainly because our PPMR dataset cannot represent the whole distribution of PMG features.

Several features of the pediatric PMG MRI dataset (PPMR dataset) are pertinent to the novel center-based deep contrastive metric (cDCM) loss function proposed. The normal brain MRI are all relatively uniform, but the PMG MRI vary among themselves in the degree of cortical thickness, irregularity, and disorganization. This is in part due to the unavoidable heterogeneity of the disease itself and partly to the limited number of available cases. After splitting the dataset into training and validation data, each subset may contain different varieties of PMG introducing bias. Anomaly detection has the potential to mitigate this issue. Moreover, the images are labelled by one senior radiology doctor and appear consistent with binary classification tasks. The combination of anomaly detection and binary classification, therefore, seems advantageous.

This motivates the design of our novel cDCM loss function. Based on our novel cDCM loss function, we customize a deep learning structure which can map each input MRI to one latent representation. And then, this loss function can "cluster" normal samples together, because normal images share similar features, and push anomaly samples further away from the "cluster". Ideally, PMG images with different pathology features may go outside of the "cluster" in different directions. During inference, according to the idea of supervised

contrastive learning [12], samples in the testing data which have similar features as normal samples in the training data can be classified as normal; on the other hand, samples in the testing data which have dissimilar features as normal samples in the training data can be classified as anomalies. Hence, there exists a center in the latent representation around which normal samples cluster, and a margin away from the center beyond which PMG samples are mapped to. This structure is created with the help of our novel cDCM loss function. The margin will also act as the decision boundary in our approach. In this thesis, we demonstrate that the final performance will not be much affected if the two hyper-parameters of center and margin are randomly set in a reasonable manner. All in all, our novel method is a binary classification method, but the idea is partly borrowed from anomaly detection and one-class supervised classification.

Overall, to deal with the challenge of detecting PMG in pediatric brain MRI, we propose a novel cDCM loss function and a custom deep learning model structure, combining dilated convolutions [13], squeeze-and-excitation [14] blocks, and a multi-level feature fusion [15] mechanism. Our novel cDCM loss function can partly mitigate the problem of generalization in deep learning, and our custom deep learning model structure can learn and select key features of MRI images automatically. In this thesis, we compare our novel cDCM loss function to the Deep SAD loss function [16] and cross-entropy-based loss functions to demonstrate the benefits of our novel cDCM loss function. Our novel cDCM loss function can directly provide a good decision threshold that cannot be determined by training and validation data. Often, the decision threshold can be determined through cross-validation on the validation data. However, since distributions from training, validation and testing data are different in our tasks, the selected decision threshold from the validation data may

not be suitable for the testing data. Additionally, based on our novel cDCM loss function, we conduct an ablation study to confirm the significance of each component in the overall structure and show the utility and stability of our custom deep learning model structure, compared to some popular CNN backbones, such as EfficientNet and ResNet50.

If children's PMG disease can be detected faster, these patients can receive early treatments. Our method can predict children's PMG medical imaging with high recall and acceptable precision. In addition, our method is also promising to be adapted to other medical imaging challenges which have small and imbalanced datasets.

1.3 Main Contributions

In summary, the main contributions of our research are listed below:

1. We propose a novel center-based deep contrastive metric learning loss function (cDCM) to partly deal with deep learning generalization problems working with a non-representative dataset. In particular, our method is able to work with a small and imbalanced dataset for which some testing data features do not appear in the training data.

2. There exists a center in our cDCM loss function, and the loss pushes normal samples close to the center while it also pushes anomaly samples away from the center. We theoretically and experimentally prove that the center with our cDCM loss can be assigned randomly.

3. To our best knowledge, we are the first to bring deep contrastive learning into the research area of PMG MRI classification. In addition, our method integrates the strengths

of supervised binary classification and anomaly detection.

4. We design a custom convolutional neural network (CNN) structure which combines dilated convolutions [13], squeeze-and-excitation [14] blocks, and a multi-level feature fusion [15] mechanism. In addition, our custom structure shows its utility compared to popular CNN models in our experiments.

5. We test our loss function on two datasets. We use a modified CIFAR-10 dataset for development and for repeatable experimentation. Our target is a pediatric PMG MRI dataset (a.k.a. PPMR dataset) from the Children's Hospital of Eastern Ontario (CHEO), Ottawa, Canada.

6. We make our PPMR dataset publicly accessible for research purposes.

1.4 Thesis structure

There are seven chapters in this thesis. Apart from this introduction, we summarize each chapter below.

In Chapter 2, we introduce important concepts used in this thesis. In Chapter 3, we give a literature review, including machine learning in PMG images, classification in medical imaging, anomaly detection in medical imaging, and deep metric learning and deep contrastive learning. In Chapter 4, we give a detailed illustration of our proposed method, including the novel cDCM loss function, mathematical proof, and custom model structure. In Chapter 5, we introduce two datasets: a modified CIFAR-10 dataset and our PPMR dataset, and also describe training details. In Chapter 6, we show the results of

substantial experiments on these two datasets and also analyze and discuss the results of these experiments. In the final Chapter 7, we give a conclusion of our thesis research and also consider some future works.

Chapter 2

Related Concepts

2.1 MRI

MRI technology is used for disease detection in radiology [17]. MRI scanners are suitable for brain images because they can differentiate gray matter from white matter, which are two important features to diagnose polymicrogyria (PMG). In addition, MRI scanners usually produce three-dimensional anatomical images, and these 3D images are typically viewed from three different directions: sagittal (from left to right), coronal (from front to back), and axial (from top to down). In Figure 2.1, the yellow plane represents the coronal plane; the red plane represents the axial plane; and the green plane represents the sagittal plane.

The interval between successive pulse sequences delivered to the same slice is known as the Repetition Time (TR). The period of time between the Radio Frequency (RF) pulse

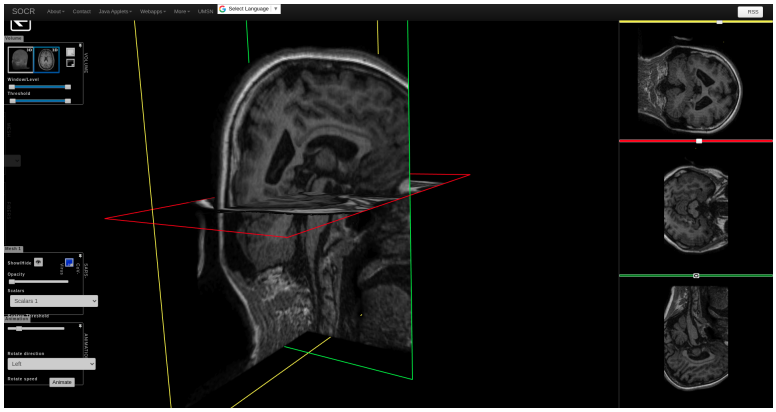


Figure 2.1: Pediatric brain anatomical planes

being delivered and the echo signal being received is known as the Time to Echo (TE). T1-weighted and T2-weighted scans are commonly used MRI sequences. Short TE and TR times are used to create T1-weighted images, while longer TE and TR times are used to create T2-weighted images. In our research, our PMG MRIs are T1-weighted.

2.2 Deep learning

Deep learning is a machine learning technique that uses representation learning and artificial neural networks (ANN). Many deep learning models can be represented as $y = f(x, \theta)$, which maps an input x to y with the help of parameters θ, where y can be a category in classification, a mask in image segmentation, or other types of representation. The parameters θ can be updated using the back-propagation algorithm based on an objective function [18]. In this section, we briefly introduce concepts and terminology about deep learning. Because we mainly use a multi-layer perceptron (MLP), convolutional neural networks (CNN), and an autoencoder in our research, more details about these neural

network architectures are given below.

2.2.1 Multi-layer perceptron (MLP)

A multi-layer perceptron (MLP), a feedforward neural network, consists of one or multiple layers of perceptrons. A MLP includes at least three layers: an input layer, one or more hidden layers, and an output layer. Each node of hidden and output layers uses a nonlinear activation function to increase the complexity of the deep learning model. The processing of each layer can be expressed as

$$X_{i+1} = \sigma(W * X_i + b) \ , \tag{2.1}$$

where W represents parameters, b is the bias term, σ is the activation function, X_i is the input of the current layer, and X_{i+1} is the output of the current layer.

Activation function. The activation function adds a non-linearity in deep neural networks. The non-linearity helps the hierarchical design of neural networks because without non-linearities several continuous linear functions can be summarized in one linear function. Deeper neural networks can solve some nontrivial problems, such as image classification and so on. ReLU is a commonly used activation function, it can be written as $f = max(0, x)$ (see Figure 2.2). Leaky ReLU is a variant of the ReLU activation function, see Figure 2.3. Leaky ReLU is defined as $f = max(\alpha * x, x)$, where α is in range from 0 to 1 and determines the slope for negative values.

The vanishing gradient problem in machine learning occurs during the training of ar-

tificial neural networks using gradient-based learning techniques and backpropagation. In such methods, each neural network's weight is updated proportionally to the partial derivative of the error function with respect to the current weight throughout each iteration of training [19]. The gradient may occasionally be so small as to be vanishingly small, which makes it impossible for the weight to change its value [19]. In the worst situation, this might prevent the neural network from receiving any more training [19]. Leaky ReLU has a small gradient when the input value is smaller than zero, and it partially deals with vanishing gradient problems.

In machine learning, a modelling error called overfitting happens when a model is too closely matched to a small number of data points. Because of this, the model is only helpful in relation to its original data set but not in relation to any other data sets. It often happens when training a complex model on a small dataset. Leaky ReLU increases the piece-wise linearity of neural networks, which reduces the complexity of the model. So Leaky ReLU can slightly reduce overfitting, compared to ReLU.

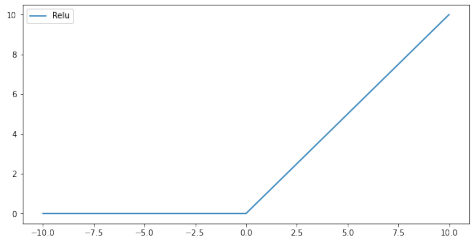


Figure 2.2: ReLU activation function

Dropout. Dropout [20] is a regularization method and it can reduce overfitting. During training, the dropout mechanism randomly drops out nodes of a neural networks. It simulates an ensemble of multiple different network structures.

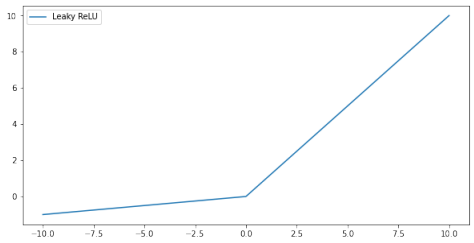


Figure 2.3: Leaky ReLU activation function

Batch normalization. In deep learning, mini-batch refers to equally sized subsets of the dataset that are used to calculate the gradient and update the weights. Since deep learning models need different mini-batch data for each iteration of training, the covariate shift often happens, which is the term used to describe the alteration in the input data distribution found in both the training and test data [21]. Batch normalization [22] makes neural networks training faster and more stable. This method normalizes and then shifts the outputs of each layer in a mini-batch to mitigate internal covariate shift, which is caused by the randomness in the input data and parameter initialization. Hence, it adjusts the distribution of the data in each mini-batch during training. It also can help to reduce the occurrence of exploding and vanishing gradients even in cases where the learning rate is high. In addition, batch normalization also acts as a regularizer, which can reduce overfitting during training.

Loss function. Loss function is a mathematical optimization function. Deep learning models update parameters to seek minima of the loss function. Commonly used loss functions are: cross-entropy (CE) loss function for classification, and mean square error (MSE) loss function for regression.

2.2.2 Convolutional neural network (CNN)

Convolutional neural networks (CNN) are commonly used in computer vision, such as image classification [23], object detection [24], etc. They replace fully-connected layers in MLP (see Sec. 2.2.1) with convolution layers and pooling layers. CNNs have the property of shift invariance. Shift invariance means differences in pixel placement, i.e., spatial shift in images, do not affect the performance because convolution kernels share weights. In addition, shared-weight kernels need less parameters, compared to MLP and hence CNNs also have a regularization property, which can reduce overfitting. The main components of CNN are introduced below.

Convolution. Convolution is a mathematical operation and it can extract features from images, such as edges. The Laplacian convolution kernel and the Sobel convolution kernel are commonly used to extract edge features. The Laplacian convolution kernel is given in Eqn. 2.2. The Sobel convolution kernel in the x-direction in Eqn. 2.3 produces a large absolute value for vertical edges, while Sobel kernel in Eqn. 2.4 has large absolute output for horizontal edges. Processed images after these three kernels are shown in Figure 2.4 and Figure 2.5. Single convolution only can extract these shallow features. However, convolutional layers convolve the input several times with the help of a non-linear activation function and pooling layers to extract more context features. Extracted features after passing convolutional layers are also called feature maps or activation maps.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.2}$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{2.3}$$

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.4}$$

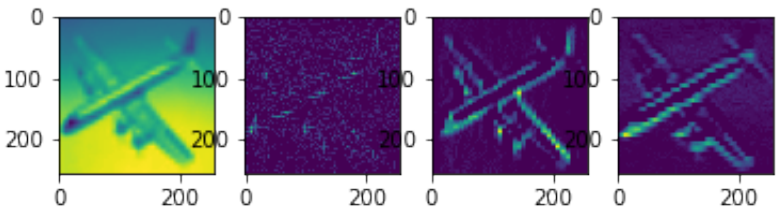


Figure 2.4: Convolution kernel processing on a CIFAR-10 airplane image. From left to right: original image, after Laplacian kernel, after Sobel-x kernel, and after Sobel-y kernel

Dilated convolution. Dilated convolution [13], sometimes also referred to as a-trous [25], is a type of convolution. It expands the receptive field by inserting holes between kernel elements. Dilation rate is a hyper-parameter to control the range of the receptive field. While convolution with a small kernel size can only extract local features, dilated convolution can extract relatively global features without increasing the number of param-

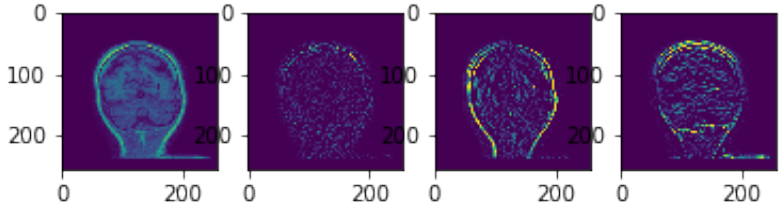


Figure 2.5: Convolution kernel processing on a our hospital PMG image. From left to right: original image, after Laplacian kernel, after Sobel-x kernel, and after Sobel-y kernel

eters. Figure 2.6 shows the difference in receptive fields between convolution and dilated convolution.

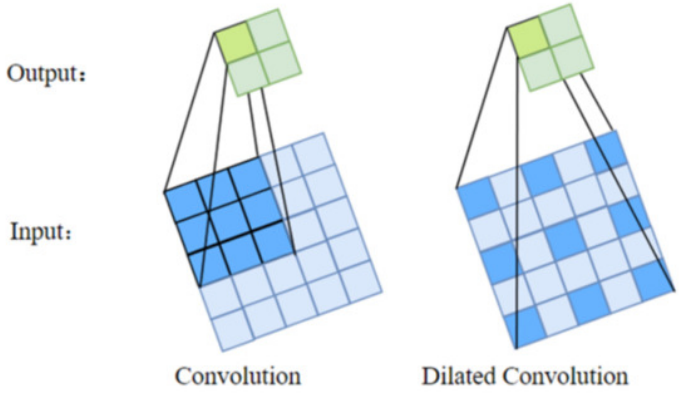


Figure 2.6: Convolution and dilated convolution, where the dilation rate is 1. [1] ©Copyright Li et al.

Pooling layers. Pooling layers in CNN reduce the size of feature maps. Although some image features information is lost after pooling layers, most prominent image features information is kept. Max pooling and global average pooling are two commonly used pooling methods. Max pooling selects the maximum value from the given region (see Figure 2.7). Global average pooling (GAP) layers average all values in each feature map to a single number (see Figure 2.8). GAP can convert 2D feature maps to a 1D vector for

further processing.

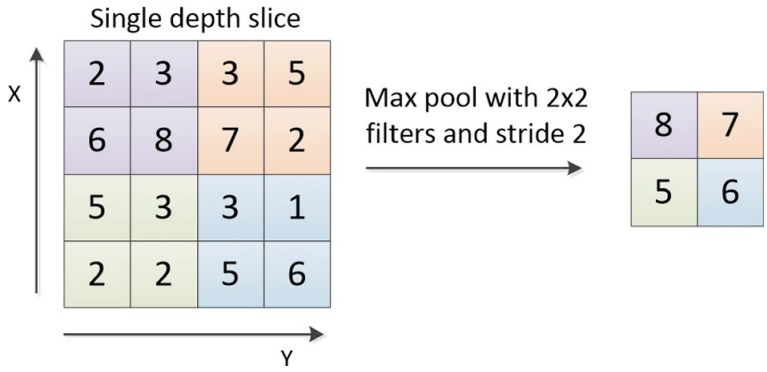


Figure 2.7: Max pooling. [2] ©Copyright Qiu et al.

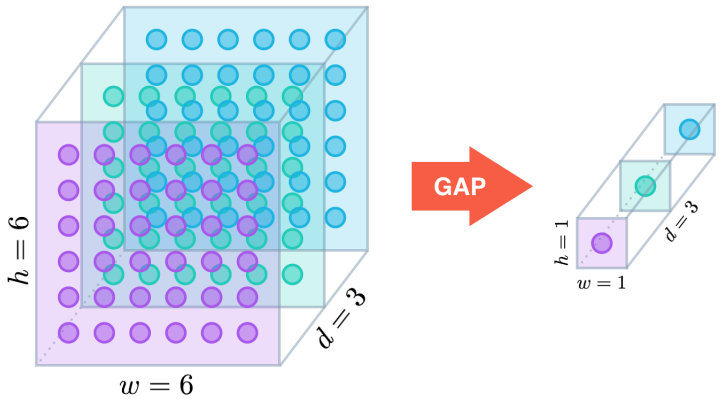


Figure 2.8: Global average pooling [3]. ©Copyright IEEE 2018.

2.2.3 CNN architectures in this thesis.

There are various CNN architectures customized for different tasks in computer vision. Several CNN backbones are used for feature extraction. ResNet, EfficientNet are selected in this thesis for CNN architecture comparison.

ResNet. Residual neural network (ResNet) [4] uses skip connections to avoid vanishing gradients. Adding skip connections can increase the depth of neural networks and achieve higher performance, because it mitigates the accuracy saturation problem. The basic unit of ResNet is shown in Figure 2.9.

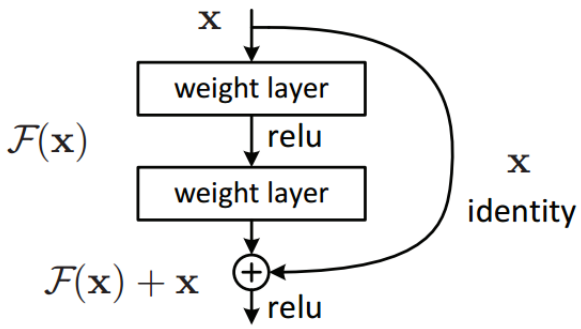


Figure 2.9: ResNet unit [4]. ©Copyright IEEE/CVF 2016.

EfficientNet. EfficientNet [5] uses neural architecture search (NAS) [26] to explore a neural network which has the optimal network depth, width and resolution to achieve high accuracy with high efficiency (see Figure 2.10). Specifically, the depth of a neural network represents the number of neural network layers; the width represents the channel dimension of each layer, and the resolution represents the size of the input feature maps fed into architecture blocks.

2.2.4 Autoencoder

An autoencoder [27] is a type of neural network. There are different kinds of autoencoders and they do very different things, such as dimensionality reduction [28], image compression [29], image denoising [30], machine translation [31], etc. One typical usage of the

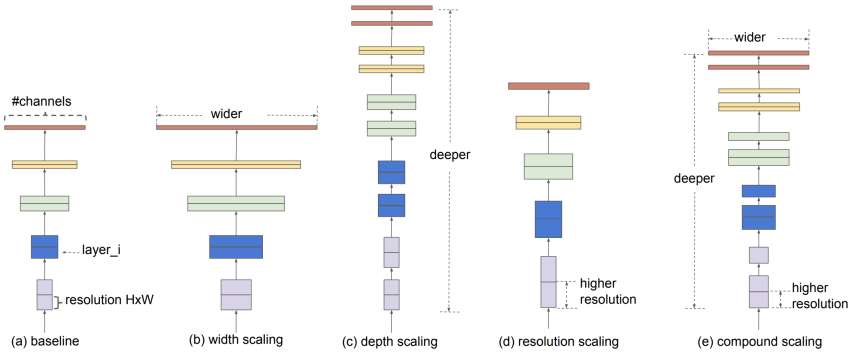


Figure 2.10: Depth, width and resolution of the EfficientNet [5]. ©Copyright Tan et al.

autoencoder is that it can learn a representation encoding a given unlabeled input data. If the representation has lower dimensionality than the input, it is an unsupervised learning method to compress data without too much information loss. The architecture of an autoencoder is shown in Figure 2.11. An autoencoder is trained to minimize reconstruction loss, where the mean squared error (MSE) loss is a commonly used loss function. The loss function is given in Formula 2.5.

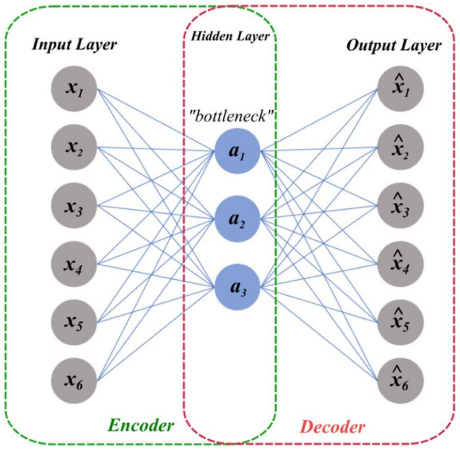


Figure 2.11: Autoencoder architecture [6]. ©Copyright Sagheer et al.

$$L_{MSE} = ||X - X'||^2 = ||X - \sigma'(W' * (\sigma(W * X + b)) + b')||^2 \tag{2.5}$$

where X is the input data; X' is the output of the autoencoder; W represents parameters of the encoder; W' represents parameters of the decoder; b is the bias term of the encoder, b' the bias term of the decoder; σ is the activation function of the encoder; σ' is the activation function of the decoder.

If the training of such an autoencoder has converged, the encoder can be used to map input data into a low-dimensional vector for further usage.

2.2.5 Neural network training strategy

Training neural networks is hard [32]. A standard training strategy is likely to find poor solutions if the task is difficult, but there are many training strategies to achieve a good solution, such as data augmentation, learning rate scheduler, early stopping, etc. Some training-related concepts and training strategies are explained in the following.

Parameters and hyper-parameters. Model parameters are denoted as θ and these can be updated during training of a deep model. Hyper-parameters are manually set before training; for example, batch size, learning rate, etc. are hyper-parameters. In addition, hyper-parameters can also be the result of a grid search or randomly sampled.

Dataset split. A dataset is normally split into training, validation and testing data. The training data are used to fit the deep learning model; validation data are used for hyper-parameter fine-tuning and model selection; and testing data are used to evaluate

the model performance. Samples from training, validation and testing data should be exclusive to avoid data leakage.

Data leakage. Data leakage means there exists data overlap or duplicates between training and validation data, or between training and testing data, or between validation and testing data. In another word, any two data between training, validation and testing sharing the same or similar data samples is called data leakage. If the dataset split is not appropriate, causing data leakage, the model performance score could be extremely high during testing but the model may still not work in practice. One should avoid data leakage as much as possible.

Imbalanced dataset. A dataset that has an unequal distribution of classes is called an imbalanced dataset. An imbalanced dataset is common in medical image classification challenges because the majority of patients are normal and a limited amount of patients have diseases. The imbalanced dataset problem can normally be solved in several ways. One possibility is to add more real data or synthesizing more meaningful data of minority classes. Another possibility, is to modify the loss function, such as using a weighted cross-entropy loss, to force the deep learning model to focus more on the minority data during training.

Out of distribution data (OOD). The data distribution in the real world could be different from the distribution of training data. The real-world data which is unseen in training data is out of distribution. OOD detection is crucial [33], especially in medical imaging diagnosis, because unseen pathology characters are very common in the real world and these unseen diseases should be detected. Hence, it is required that deep learning

models need good generalizations on unseen data.

Model generalization and overfitting. Model generalization refers to how well is the model able to generalize on unseen data if these unseen data are from the same distribution of the training data. Overfitting means the model generalization is bad because the training performance is good, but the validation or testing performance is much lower. Data augmentation, learning rate scheduler, and early stopping are commonly used to mitigate the problem of overfitting during training.

Data augmentation. Accessing sufficient labeled data in the real world is hard. In order to improve the quantity of training data, data augmentation is an inevitable part of training deep learning models [34]. Data augmentation techniques modify the original data slightly to increase the amount of data. It is a kind of regularization method because it can reduce overfitting during training and increase the robustness of deep learning models. Data augmentation in deep learning is related to oversampling in conventional machine learning. Rotation, shift, shearing and zooming are commonly used imaging data augmentation methods. Deep generative models, mixup [35], and cutmix [36] are also used for synthetic data generation.

Mixup is a data augmentation method which blends two images into a new image. Mixup can be represented in Formula 2.6 and 2.7.

$$\hat{x} = \lambda * x_i + (1 - \lambda) * x_j \tag{2.6}$$

$$\hat{y} = \lambda * y_i + (1 - \lambda) * y_j \tag{2.7}$$

where x represents the image; y represents the label; and λ controls the ratio between two images. It is also worth noting that λ is sampled from the Beta distribution with the range from 0 to 1. As an example, Figure 2.12 represents two original images before mixup, while Figure 2.13 represents the new image after mixup.

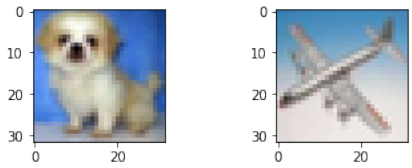


Figure 2.12: Two original images

Figure 2.13: New image after mixup

Cutout [37] data augmentation removes pixels from the image and fills them with meaningless color, such as black or gray. Figure 2.14 demonstrates two new images after cutout. However, cutout could lead to information loss of the original images. Cutmix is similar to mixup and it combines mixup and cutout. Cutmix removes pixels from one image but replaces them with another image, and the label of the new image is interpolated by the labels of these two images. Cutmix achieves better performance on CIFAR and ImageNet classification tasks, compared to mixup and cutout [36]. Figure 2.15 shows two new images after cutmix.

Reduce learning rate on the plateau. A large learning rate may miss the global optimum, while a small learning rate may get stuck in a local optimum. E.g., "reduce

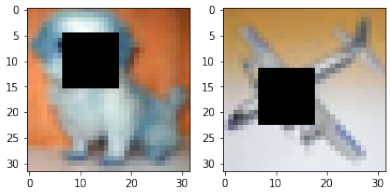


Figure 2.14: New images after cutout

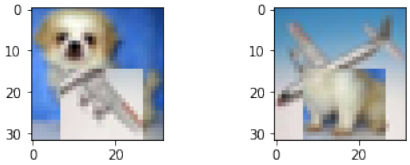


Figure 2.15: New images after cutmix

learning rate on the plateau" is a learning rate scheduler. It does not require much effort in hyper-parameter tuning. When a prearranged metric stops improving within the limits of patience, the learning rate should be reduced and the model continues to find a "more global" optimum.

Early stopping. Normally, an early stopping mechanism stops neural network training when the loss no longer decreases on validation data to avoid overfitting. In our research, we use a specific training method which including early stopping. More details can be found in Section 5.3 Experimental Settings.

Transfer learning. Transfer learning [38] improves one learning task based on the knowledge learned from a related task. In the research area of computer vision, especially in convolutional neural networks (CNN), the extractor of shallow features, such as edges, can be pre-trained on a relatively large dataset, such as ImageNet. And then this pre-trained feature extractor can be adapted to a new dataset which may have not sufficient

data to train a deep learning model from scratch. Transfer learning has been shown to improve results on various tasks [39, 40]. Typical steps during transfer learning are:

- Creating a model and then training this model on a relatively large dataset.
- Freezing layers of the pre-trained model backbone and adding new trainable layers on the head.
- Training new layers on the target dataset.
- Unfreezing and fine-tuning all layers on the target dataset with a relatively low learning rate.

2.3 Summary

In the beginning of this chapter, we described concepts related to MRI because of our PPMR dataset. Then we gave a brief overview of some deep learning concepts. Of these concepts, loss function, dilated convolution, imbalanced dataset, out-of-distribution data, model generalization and overfitting are especially important throughout the whole thesis. We introduce related work in the literature review of the next chapter.

Chapter 3

Related Work

In this chapter, we give a literature review, which is related to our proposed method in this thesis. Disease identification in MRIs is typically accomplished using classification [41] and anomaly detection techniques [42]. In Section 3.1, we introduce some work applying machine learning to images containing polymicrogyria (PMG). In Section 3.2, we introduce classification methods to detect a disease from medical images, which include traditional machine learning methods, deep learning methods, and some combinations of these two methods. In Section 3.3, we introduce anomaly detection methods to detect unseen diseases from medical images. However, both previous binary classification methods and anomaly detection methods do not work well on our PPMR dataset. In Section 3.4, we introduce the research area of deep metric learning or deep contrastive learning, because our method is inspired by these methods. As we are the first, to the best of our knowledge, to bring deep contrastive learning methods to PMG detection based on MRIs, there are no direct comparison methods.

3.1 Machine Learning in Polymicrogyria Images

There is limited literature about applying machine learning methods on detecting pediatric PMG from MRIs [43]. Attallah et al. [43] proposed a machine learning pipeline for fetal brain MRI classification. This method can predict several brain abnormalities, including PMG, before the infant with fatal brain abnormalities is born. This machine learning pipeline consists of four steps: segmentation of fetal areas on the whole brain to get the region of interest (ROI), contrast enhancement on the minor ROI image, feature extraction by discrete wavelet transform (DWT) or other statistical methods, and classification by linear discriminate analysis (LDA), support vector machine (SVM), K-nearest neighbour (KNN) and ensemble classifiers. The best result achieves 80% accuracy, 84.5% AUC, 81.4% sensitivity, and 83% specificity. However, feature extraction of these methods is manually designed and has downsides. Moreover, we do not have the segmentation annotation in PPMR dataset, so the first step of this method is not suitable for our task. Attallah et al. [9] continued to propose a method which combines CNN feature extraction based on transfer learning and conventional machine learning classifiers to detect fatal brain MRIs, and this method achieved 88.6% accuracy and 94% AUC. We do not compare our method to Attallah et al. [9], despite the fact that it performed well on MRIs, because transfer learning did not extract useful features from our PMG images in our tests(see Section 6.7), which differ only slightly from control images. Hence, we could not apply the approach of Attallah et al. [9] as transfer learning is required.

In addition, Plonski et al. [44] collected 740 features, such as cortical thickness and surface area, from T1-weighted MRIs and then used feature selection methods and a logistic

regression classifier to detect congenital brain malformations including PMG that lead to dyslexia. The best results are 66% AUC and 65% accuracy after 10-fold cross-validation. They demonstrated the main reason leading to these relatively poor results is that dyslexia is a heterogeneous syndrome and there is little consistency in univariate grey matter.

However, these methods above are not specific to the detection of PMG on MRIs, because they applied their methods to a dataset which only includes a small portion of PMG. In addition, these methods have not shown they have the ability to deal with the problem of imbalanced datasets and generalization to unseen features.

3.2 Classification in Medical Imaging

In medical imaging, binary classification is often applied to decide whether an image includes disease features, and then assist doctors in diagnosing if patients have some specific diseases during routine clinical flow [45–47]. Medical imaging classification tasks can be divided into three main categories: conventional machine learning methods, deep learning methods, and combinations of the two.

Conventional machine learning methods for medical image classification tasks usually need several steps to implement the whole method processing pipeline: 1. image pre-processing if needed 2. image feature extraction 3. feature selection or transformation 4. machine learning models training and evaluation. Image pre-processing includes pixel values normalization [48], tile correction [49], noise removal [50], skull stripping [51], etc. Some popular feature extraction methods are edge detection, image filtering and enhance-

ment, region of interest (ROI) identification, etc. [52]. After feature extraction from various methods, there are plenty of different features.

The "Curse of Dimensionality" was initially introduced when considering the problem of dynamic programming [53], which refers to various phenomena. In the research field of machine learning the "Curse of Dimensionality" is a phenomenon that when the amount of data samples is fixed, the predictive ability of a classifier increases with increasing the number of features or dimensions until a certain dimensionality, and then the predictive ability of the classifier decreases with increasing the number of features or dimensions [54]. Conventional machine learning models only accept input features with limited dimensions to avoid the "Curse of Dimensionality" [55]. Hence, feature selection or transformation are applied to reduce the dimensionality of feature vectors. For example, principal component analysis (PCA) [56], singular value decomposition (SVD) [57], linear discriminant analysis (LDA) [58] are commonly used for projecting features into lower dimensions. Conventional machine learning models, for example, support vector machines (SVM) [59], decision trees [60], etc., are trained based on these processed features for classification tasks. For examples, Zhou et al. [61] used luminance grade of X-ray computed tomography (CT) images as the features and applied a decision tree model as a classifier for computer-aided diagnoses. Lo and Wang [62] extracted spectral signatures from breast multi-spectral magnetic resonance images and applied a support vector machine model to these features for breast cancer detection. Othman et al. [63] obtained features of magnetic resonance imaging (MRI) images using discrete wavelet transformation and applied an SVM model for brain disease classification. Alhindi et al. [64] used features extracted from histopathology images using local binary patterns (LBP) and applied an SVM classifier. Le et al. [65]

proposed a new SVM method for medical image classification.

However, conventional machine learning methods have some limitations. They need handcrafted feature engineering [66], and essential features can be difficult to extract or extracted features may not be optimal. Researchers may not be able to translate disease patterns on medical images into decent feature descriptors via traditional image processing methods [67], mainly because of the lack of medical background or the lack of knowledge in image processing. Poor quality inputs will cause training failure or unreliable results on conventional machine learning models [68], so these models cannot classify images correctly or with decent results. In contrast, deep learning methods can learn high-level features from input data samples directly [67], and these features are extracted to meet the goal of the task by the designed loss function and the back-propagation algorithm. With high volume data and increasing high-performance hardware, especially GPUs, deep learning methods possess great advantages over conventional machine learning methods. Li et al. [23] customized a convolutional neural network (CNN) framework with shallow layers to classify diseases on lung image patches. Wang et al. [69] tailored a CNN model, named as COVID-Net, to detect COVID-19 disease from chest X-ray images. In addition, Yadav and Shivajirao [70] compared CNN models, including VGG16 and InceptionV3, and the SVM classifier on chest X-ray images to classify pneumonia and demonstrated CNN models can achieve better performance than SVM. Apart from CNN structures, transformer-based structures have also been used in medical imaging classification tasks in recent years [71–74]. Dai et al. [71] applied a transformer model, called TransMed, on multi-modal medical images to classify parotid gland tumors and knee injury. But transformer-based models require large-scale datasets for training [71], and, therefore, this kind of structure is not

suitable for our task.

Medical brain data is typically volumetric. Since 3D brain MRIs include more information compared to 2D brain MRIs because of the third dimension, some researchers applied deep learning models to 3D brain MRIs directly and achieved good results when they have sufficient 3D volumetric data [75, 76]. Korolev et al. [75] proposed an end-to-end 3D CNN architecture to classify Alzheimer's disease from normal controls on the 3D Alzheimers Disease National Initiative (ADNI) dataset. Wegmayr et al. collected 3D brain MRIs from various sources to build a large 3D brain dataset and customized a 3D CNN architecture to classify several neurodegenerative diseases [76]. Mehrtash et al. used a 3D CNN architecture on two 3D MRI modalities to detect prostate cancer [77]. However, in our PPMR dataset, we only have 23 PMG brains in total, which is not sufficient for training a 3D CNN model. We can regard each PMG brain as a series of PMG slices; then the number of data samples is sufficient to train a deep learning model.

In medical imaging, it is difficult to acquire a sufficient number of images because of a limited number of patients and because of privacy concerns [78]. Deep learning models are easily overfitting to the training data if they are trained on relatively small datasets [79]. Consequently, deep learning models will lack the ability to generalize on unseen medical images or unseen pathological characterizations. To avoid shortcomings of conventional machine learning methods and deep learning methods, some researchers regard deep learning methods as feature extractors and then apply these extracted features to conventional machine learning models. They demonstrated this combination could partly improve generalization [39]. Commonly used feature extractors are popular convolutional neural networks (CNN), because CNN models can share shallow features from various datasets, which is a

form of transfer learning [80]. These CNN feature extractors need to be first pre-trained on a larger dataset, for example, the ImageNet Dataset is commonly used, and then fine-tuned on the specific task dataset. Hence, these combined methods normally have a two-stage or a multi-stage training process. Alinsaif and Lang [81] extracted features from medical images using pre-trained CNN models and then these feature vectors were fed into a SVM. This method was tested on four different medical imaging datasets, leading to higher accuracy than previously published results. Liu et al. [82] fused handcrafted features and deep features extracted from pre-trained CNN models with transfer learning, used a ReliefF algorithm [83] for feature selection, and applied SVM to these selected feature vectors for medical image classification tasks.

However, in our case, deep learning models with the help of transfer learning and traditional image processing methods cannot extract essential features from our hospital PMG images, because the differences between normal and PMG MRI are subtle. Because of it, we develop a one-stage end-to-end deep learning method rather than a multi-stage method. In addition, we plan to collect more PMG data in the future. Theoretically, all machine learning models can achieve better performance by increasing the amount of training data. Moreover, deep learning methods show advantages with a large amount of data than traditional machine learning methods [84].

3.3 Anomaly Detection in Medical Imaging

Pathologies in medical images are often regarded as rare deviance (anomalies) because the majority of medical images are healthy (or normal) samples [42, 85]. Anomaly detection in

medical imaging is a research area where these anomalies can be detected or selected automatically. Unlike binary classification tasks, anomaly detection can deal with a relatively imbalanced dataset where normal samples are the majority and the diversity of anomaly samples is limited.

In conventional anomaly detection methods, first, features on pre-processed medical images may be extracted by traditional image processing methods. Secondly, principal component analysis (PCA) or other above-mentioned feature selection methods may be applied to the extracted features to avoid the phenomenon of the "Curse of Dimensionality". Lastly, statistical methods, i.e., z-score, density analysis, etc. [42], can be used to detect anomalies. Additionally, some one-class conventional machine learning models also work well on anomaly detection, i.e. one-class support vector machine (OCSVM) [86], support vector data description (SVDD) [87], etc. Zhang et al. [88] extracted features using completed local binary patterns (CLBPs), gray level co-occurrence matrix (GLCM), and curvelet transform from breast cancer biopsy images and applied a one-class kernel principle component analysis (KPCA) model to these extracted features. Wang et al. [89] presented a Python package, called OCTID (one-class tumor image detection). This tool includes three components in the whole processing pipeline: extracting features from pre-trained CNN models; uniform manifold approximation and projection (UMAP) for dimension reduction, and OCSVM for classification. In addition, Gao et al. [90] applied OCSVM on highly imbalanced medical imaging datasets.

Deep learning based anomaly detection methods are divided into supervised, semi-supervised and unsupervised learning methods. Semi-supervised and unsupervised methods are commonly used in anomaly detection because collecting ground truth labels with

high confidence is often time-consuming and costly [42]. Supervised anomaly detection methods use well-labeled imbalanced datasets for training and focus more on detecting anomaly samples. Semi-supervised anomaly detection methods only select pure normal samples to train, ignoring data samples which have anomaly labels or which have no explicit labels. Unsupervised anomaly detection methods use all unlabeled data samples for training, and this unlabeled dataset could include both normal samples and anomaly samples. Seebock et al. [91] trained a deep convolutional autoencoder (DCAE) on healthy retinal images and used extracted latent vectors to train an OCSVM. Tlusty et al. [92] trained "stacked" denoising autoencoders [93] in which multiple autoencoders are stacked but the output of each layer in the autoencoder is randomly corrupted by a dropout layer. They applied a K-means clustering algorithm on extracted latent vectors to detect anomalies. Tang et al. [94] trained deep convolutional generative adversarial networks (DCGAN) only on normal chest X-ray images. The well-trained DCGAN can reconstruct normal images well but fails to reconstruct anomaly images. In addition, Watanabe et al. [95] trained an AnoGAN model only on normal computed tomography (CT) images to detect bone metastatic tumors. However, semi-supervised and unsupervised methods may be limited for medical images which have complex or even indistinguishable features [96], such as PMG images.

Therefore, exploring anomaly-supervisory signals may help anomaly detection [97]. Luff et al. introduced a deep semi-supervised anomaly detection (Deep SAD) [16] which showed that adding a few labeled anomaly samples during training can achieve better results than the deep support vector data description (Deep SVDD) [98], which is trained only on normal samples. Ding et al. [99] proposed a novel open-set supervised anomaly detection

method to detect seen anomalies and unseen anomalies. Their model used a few labeled anomaly samples and created some pseudo anomaly samples which are partly from outer data sources during training. The pseudo anomaly samples in their research seem to have similar features as real anomaly samples while both pseudo and real anomaly samples are dissimilar to normal samples. However, in our PMG detection research, if we create pseudo anomaly samples, real anomaly samples will be more close to normal samples because the differences between PMG MRI and normal MRI are subtle and even junior radiologists cannot diagnose them with high confidence. Hence, introducing pseudo anomaly samples for our PMG detection may not be appropriate.

Imbalanced dataset problems can be solved by binary classification methods or anomaly detection methods [96]. However, binary classification methods have generalization problems because they cannot predict images with unseen features correctly. Anomaly detection methods are limited in disentangling anomaly samples from normal samples if the differences between them are subtle because anomaly detection normally refers to the detection of anomaly samples which significantly deviate from the majority of normal samples [97]. To solve imbalanced dataset problems with good generalization, we combine the idea of binary classification and anomaly detection. Our proposed method can learn a deep learning model which can cluster normal samples together inside a hypersphere and push anomaly samples far away from the cluster center. Hence, seen and unseen anomaly samples have dissimilar features to normal samples, so they are classified as outliers of the hypersphere.

3.4 Deep Metric Learning and Deep Contrastive Learning

Deep learning methods can be trained end-to-end thanks to the back-propagation algorithm. Unlike conventional machine learning methods, which require complicated and professional feature engineering steps, deep learning models can extract features from medical images automatically without (many) image pre-processing steps. Hence, we mainly introduce recent works on deep metric learning and deep contrastive learning in the following.

Deep metric learning [66] and deep contrastive learning [12] sometimes are confused by researchers because they share similar ideas. Both of them learn embedding representations from inputs. However, deep metric learning was inspired by traditional metric learning methods [100], i.e., t-SNE [101], K-nearest neighbors [102, 103], and relevant variants. Metric learning maps input data samples into embedding representations and then calculate distances between these embeddings for clustering or classification tasks. Deep metric learning replaces sophisticated manually designed mapping functions with deep neural networks. On the other hand, deep contrastive learning is a subset of deep metric learning. Deep contrastive learning methods learn embedding representations in a contrastive fashion [104]. It requires that similar data samples (data samples from the same class) should be close to each other in the embedding space. On the other hand, dissimilar data samples (data samples from different classes) should be far away from each other in the embedding space. In another words, similar data samples should have similar embedding representations, whereas dissimilar data samples should have dissimilar ones. Han et al. [105] used a contrastive learning method to pre-train a feature extractor in a self-

supervised way and added a classification head to fine-tune the whole model for pneumonia detection on chest X-ray images. Similarly, Chen et al. [106] also used a contrastive loss to train an encoder. But they adopted the pre-trained encoder for classification using a few-shot learning method to diagnose COVID-19 based on chest CT images. More similar works [107–110] used contrastive learning methods to learn better feature representations from medical images. Deep contrastive learning methods do not always calculate distance based on the learned feature embeddings. To make the statement clearer, we call our method a deep contrastive metric learning method.

Overall, deep contrastive metric learning methods can map input images into a different latent feature space. Due to limitations of cross-entropy-based loss functions in our challenge, we use the idea of deep contrastive metric learning directly, rather than with pre-training. Based on our proposed novel loss function, a mapping function can be learned to force normal samples into a hypersphere and push anomaly samples away from this hypersphere. A prediction decision can be made directly according to the boundary of the hypersphere.

3.5 Summary

In this chapter, we introduce some machine learning methods applied to the classification of fecal brains or brain malformations, but there is no method to specifically detect PMG on MRIs. In addition, we not only introduce binary classification and anomaly detection methods but also illustrate the limitations of these two types of methods for the detection of PMG on MRIs from our PPMR dataset. Moreover, we introduce deep metric learning and

deep contrastive learning methods, which inspire our novel center-based deep contrastive learning method to detect PMG on MRIs. Our method combines the strength of binary classification and anomaly detection. In the next chapter, we introduce our proposed method in detail.

Chapter 4

Proposed Method

4.1 Overview

The diversity of polymicrogyria (PMG) MRIs is small and the number of PMG MRIs and the number of normal MRIs are imbalanced in our PPMR dataset. Cross-entropy-based loss functions have several limitations: they easily lead to overfitting during training since the dataset is small; they tend to predict more testing samples as the majority class; and they lack the generalization ability on unseen features. To mitigate these issues, we propose a novel cDCM loss function for center-based deep contrastive metric learning and a custom convolutional neural network (CNN) structure for PMG detection in pediatric brain MRIs.

Our novel cDCM loss function can improve model generalization, and the decision boundary of this loss function tends to produce high recall with acceptable precision over two datasets: the modified CIFAR-10 and the PPRM. More details about our novel cDCM

loss function are demonstrated in section 4.2. Since the goal of our research is to design a computer-aided diagnosis method to identify possible PMG, and radiologists should focus on images predicted as possible PMG, recall is more important than precision in our setting.

We customize a deep learning structure which is specifically designed for our PPMR dataset and compare our custom model with two popular CNN structures, in particular, to EfficientNet and to ResNet50. Since radiologists need to focus on both local features, such as gyri and boundary, and global features, such as spatial surroundings, our custom structure is designed to extract both local and global features. Moreover, our custom structure is composed of dilated convolutions [13], squeeze-and-excitation [14] blocks and DenseNet-like [15] feature fusion, to assist in extracting key features of PMG on MRIs. More details about our custom model structure are discussed in Section 4.3.

4.2 Loss Function

Inspired by the loss function of Deep SAD [16] and the hinge loss [111], we propose a novel center-based deep contrastive metric learning loss function (cDCM), which has a clear pre-defined decision boundary to help make the final decision. Our deep learning model f maps each input image to an n-dimensional latent representation.

Specifically, Deep SAD uses

$$Loss = \frac{1}{n}\sum_{i=1}^{n} d_i^{2*y_i} \ , \tag{4.1}$$

where $y_i = \{-1, +1\}$; $y_i = +1$ denotes normal samples; and $y_i = -1$ denotes anomaly

samples. d_i is the distance from the latent representation of sample i to the center c. Deep SAD loss can force normal samples close to the center c and push anomaly samples far away from the center c, but there is no clear decision boundary for the Deep SAD loss.

Consider the hinge loss, which is defined as

$$L_{Hinge} = max(0, 1 - y * \hat{y}) \ , \tag{4.2}$$

where y is the ground truth (either +1 or -1), and $\hat{y}$ is the prediction of the classifier.

The basic idea of our novel cDCM loss function is shown in Figure 4.1. This loss can force negative (normal) samples to be close to the center c, and it can also push positive (anomaly) samples outside the circular margin centered at c.

We use the Euclidean norm to measure the distance between latent representations of samples and the center c.

$$d_i = ||l_i - c|| \tag{4.3}$$

where l_i is the latent representation of sample i, and d_i is the Euclidean norm between the latent representation and the center c.

Our loss function is then

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(1 - y_i) * d_i + \frac{1}{M}\sum_{j=1}^{M}\alpha * y_j * max(0, m - d_j) \tag{4.4}$$

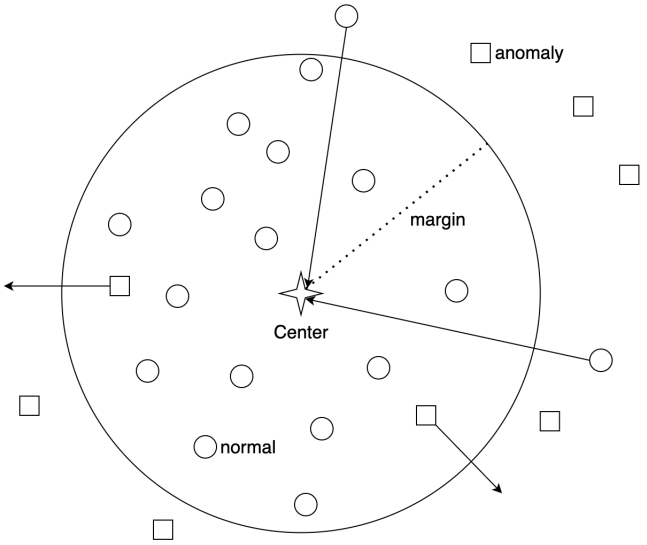


Figure 4.1: Illustration of our novel cDCM loss function, the four-pointed star represents the center c, anomaly samples in the latent representation are marked with a square, normal samples in the latent representation are represented by a small circle, and the large circle represents the decision boundary.

where N is the number of normal samples in one batch; M is the number of anomaly samples in the same batch; m is the margin, which can also be regarded as the distance from the center c to the decision boundary, and y is the label. Positive (anomaly) samples are labelled y=1, whereas y=0 is the label for negative (normal) samples.

We initially experimented with the loss function in Equation 4.4, but we found many positive (anomaly) samples distributed around the decision boundary after the training converged. To make sure the majority of positive (anomaly) samples are not close to the decision boundary (margin), we add the term $\frac{1}{1+e^{d_i-m}}$, which increases the gradients for positive (anomaly) samples which are outside of the circle but close to the decision boundary (margin). This term pushes positive (anomaly) samples further away from the

center c. We also add the parameter α to account for an imbalanced dataset and hence our novel overall loss becomes

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(1-y_i)*d_i + \frac{1}{M}\sum_{j=1}^{M}\alpha * y_j * (max(0, m-d_j) + \frac{1}{1+e^{d_j-m}}) \ . \tag{4.5}$$

The values of the different loss terms in Eqn. 4.5 depending on distance from the centre are plotted in Figure 4.2.

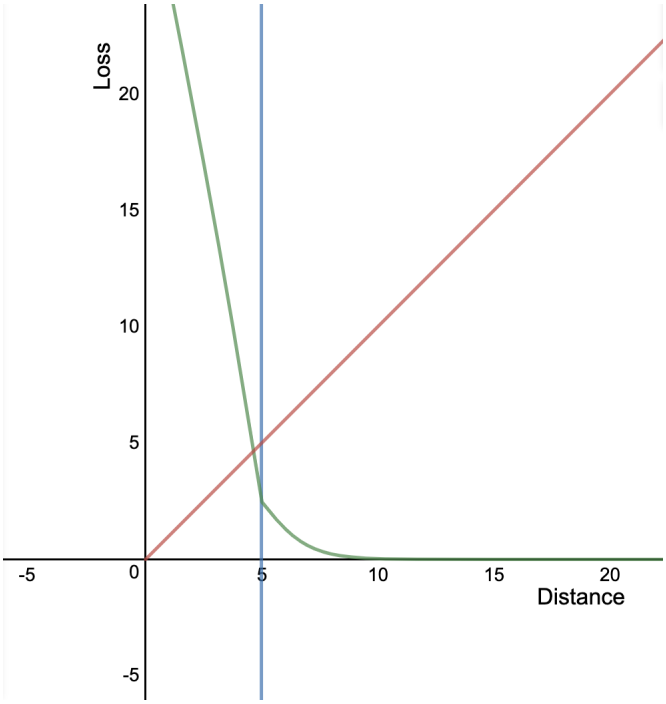


Figure 4.2: Loss value plot. The red line in the plot represents the loss of samples which are labeled as negative (normal), whereas the green line represents the loss of samples which are labeled as positive (anomaly). In addition, the blue line is the decision boundary (margin).

A center c is also employed in the deep contrastive metric learning methods: Deep SVDD [98] and Deep SAD methods. However, these methods use an autoencoder which is

trained first and then, a center c is calculated based on the average of latent representations of all normal samples in the training data. Deep SAD demonstrated fixing the center c during training can achieve smoother and faster convergence [16]. In our research, we theoretically and experimentally prove that the hyper-parameter center c can be chosen randomly at initialization time, and then be fixed during training.

Since the center c is one important component of our loss function, choosing the correct center c may appear essential. Next, we will show however, that this center c can be chosen randomly at initialization time, and experimentally verify this proposition again in Section 6.5.

Proposition: Given a specific deep learning structure, ideally, if one center c' is initialized randomly and is fixed during training, one model trained on this center c' can achieve the same optimum as other models trained on other centers. It is not necessary to find the true center c first.

Proof:

1. Assume there is a true center $c(c_0, c_1, c_2, ..., c_n)$ of the latent representations of all normal samples, and c is a n+1 dimensional vector

2. Our randomly defined center is $c'(c_0 + s_0, c_1 + s_1, c_2 + s_2, ..., c_n + s_n)$, where s_n is a constant value.

3. Assume there is an hypothetical well-trained model f based on the correct center c, so $f(x) = (l_0, l_1, l_2, ..., l_n)$. l_i represents the i^{th} value of the latent representation l in dimension i.

4. Our well-trained model based on our randomly initialized center c' can be represented as $f'(x) = (l'_0, l'_1, l'_2, ..., l'_n)$.

5. The squared distance between the latent representation of a sample in the hypothetical well-trained model f from the true center is $d^2 = (l_0 - c_0)^2 + (l_1 - c_1)^2 + (l_2 - c_2)^2 + ... + (l_n - c_n)^2$

6. And the squared distance between the latent representation of a sample in our well-trained model f' from the randomly defined center is $d'^2 = (l'_0 - c_0 - s_0)^2 + (l'_1 - c_1 - s_1)^2 + (l'_2 - c_2 - s_2)^2 + ... + (l'_n - c_n - s_n)^2$

7. If there exists a simple method to ensure that $d^2 == d'^2$ for each sample, our well-trained model f' based on center c' will have the same global loss value as the well-trained model f based on the center c. An obvious solution would be

$$\begin{aligned} l_0 &== l'_0 - s_0 \\ l_1 &== l'_1 - s_1 \\ l_2 &== l'_2 - s_2 \\ &\dots \\ l_n &== l'_n - s_n \ . \end{aligned}$$

8. We can write l'_n as $l_n + k_n$, where k_n is another constant value. If we want $l_n == l'_n - s_n$, we must ensure $l_n == l'_n + k_n - s_n$ and hence we have to set $k_n == s_n$.

9. We consider s_n as the shift of the true center c. And intuitively, k_n can be regarded

as the shift of latent representations in the latent space. A shift of the latent representations can be achieved by the bias term of the last fully-connected layer in the model. Hence, although the loss is only minimized for all training samples as a whole, there exists a simple solution for deep learning models to meet $d^2 == d'^2$ for each sample by modifying only the bias term.

10. The minimum loss value is 0 given our novel cDCM loss function. In this case, all normal samples will be located at the position of the center in the latent representation. Hence, shifting centers means shifting all normal samples in the same direction and it is not necessary to find the true center based on normal samples.

Overall, the training process could simply ensure that the the correct center c shifts to random center c', and the latent representation of each sample based on the correct center c will also be shifted, to keep the same distance from each data sample to the center. However, if the loss does not reach 0 for the normal samples, and considering the loss caused by the anomaly samples, there could be other complex transformations for these latent representations to make sure the global loss is optimal.

In addition, different model structures likely map samples differently into the latent space and hence some may be better than others. Therefore, model structure is also important to achieve good performance.

4.3 Model Structure

Although we introduce a novel cDCM loss function which can partly deal with the generalization problem on a small imbalanced dataset, it is still necessary to find a model which can extract essential features from our hospital PMG images. According to PMG disease characteristics in medical images, gray matter contour in a patient's brain and gray matter location in relation to the whole brain are both important features to distinguish the observed medical images.

We design a custom CNN structure as shown in Figure 4.3. This model is inspired by dilated convolution [13], squeeze-and-excitation (SE) [14] and a channel-wise attention mechanism during feature fusion. In each SE-Dilated-Block (see Figure 4.4), we use dilated convolution with three different dilation rates to extract features from different receptive fields. Dilated convolution with a small dilation rate (e.g. 1) can extract local features, the same as standard convolution, and dilated convolution with a large dilated rate can extract global features [14].

However, although global and local features are both important, the importance of local features varies in different layers of the model. For example, shallow layers may extract local features but deep layers may extract more global features. To deal with this issue, we use the squeeze-and-excitation block, which performs like a channel-wise attention [112], to help us select important feature maps dynamically during training. In the whole structure, shallow SE-Dilated blocks extract local features, and deep SE-Dilated blocks extract global features. Fusing shallow features and deep features may improve overall performance. A Multi-Layer Perceptron (MLP) head maps fuses features

into a low dimensional vector, since we calculate a distance metric in our novel cDCM loss function. We also note that a low dimensional latent space helps to address the "Curse of Dimensionality" when calculating distances [55]. More details about this model structure are demonstrated below.

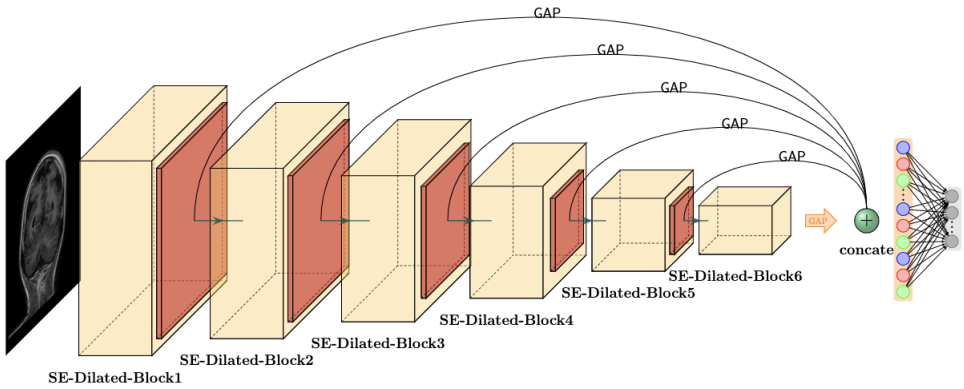


Figure 4.3: Model structure for PPMR dataset classification. GAP means global average pooling. Red rectangles represent max pooling.

4.3.1 Dilated Convolution

In each SE-Dilated-Block, we use dilated convolutions with three different dilation rates, 1, 2, and 3 separately. Dilated convolution with a dilation rate of 1 can extract local features, and with a dilation rate of 3 can extract relative global features. Three different dilated convolutions can extract local and global features together. In addition, three versions of feature maps are concatenated together for further processing.

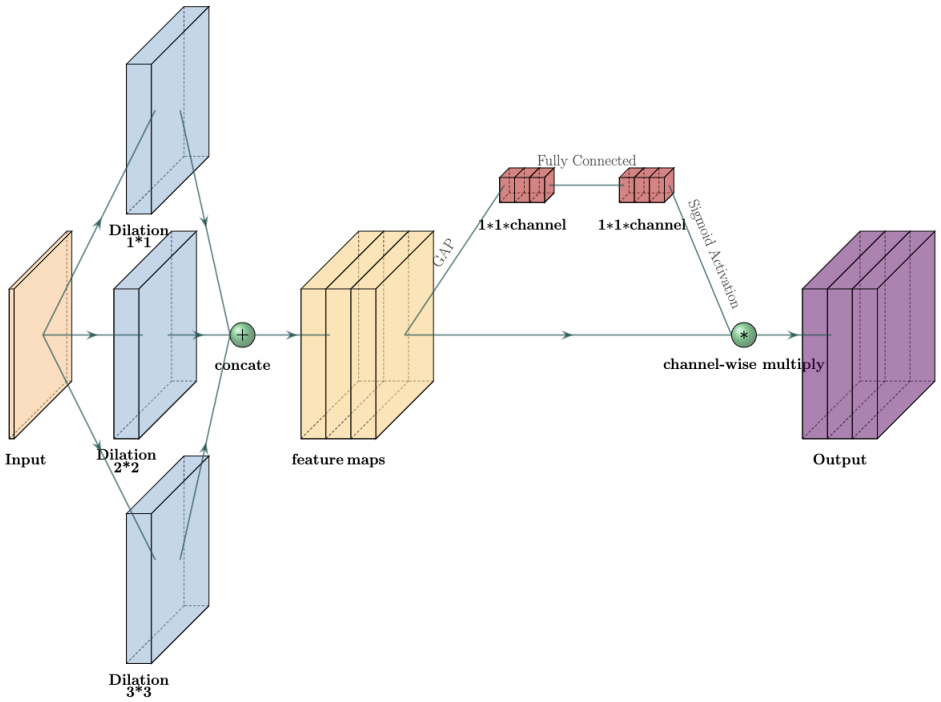


Figure 4.4: SE-Dilated-Block

4.3.2 Squeeze and Excitation Block

Intuitively, not all feature maps extracted from dilated convolutions are essential. Maybe local features are more important in some layers, but global features are more important in other layers. Squeeze-and-excitation (SE) blocks perform like a channel-wise attention mechanism [112]. Attention weights are multiplied on each feature map, and high weights should be assigned to essential feature maps, whereas low weights should be assigned to insignificant feature maps. Hence, with SE blocks, attention is automatically directed to local or global features as appropriate. Channel-wise attention weights are generated from their own feature maps. Global average pooling processes feature maps from dimension

$W \times H \times C$ to dimension $1 \times 1 \times C$. Then, an MLP layer changes its parameter, but the dimension stays the same. Finally, the sigmoid activation function transforms attention scores to a range from 0 to 1, where 0 means ignoring the feature map in a channel is appropriate; and a score close to 1 means the feature map in another channel is important.

4.3.3 Feature Fusion

Concatenation of three versions of dilated feature maps is a type of feature fusion. Apart from that, we fuse feature maps from different layers via DenseNet-like [15] connections. Since the fused feature format should be a 1D vector, global average pooling layers can transform a series of 2D feature maps to this 1D vector.

4.3.4 Dimensionality Reduction Head

Xia et al. [113] demonstrated Euclidean distance becomes less effective with increasing dimensionality. Since we calculate the Euclidean distance in our loss function, the dimensionality of the learned latent representations should not be high. In addition, these fused features need to be selected for prediction. We use several MLP layers to simultaneously, transform features and reduce the dimensionality.

4.4 Summary

In this chapter, we introduce our proposed method, including our novel cDCM loss function and custom model structure.

We illustrate the idea and formula of our novel cDCM loss function. Since there exists a center in this loss function, we give a theoretical proof to verify that this center can be assigned randomly in a proper way.

As for our custom deep learning model structure, we demonstrate our insights to design this model and introduce the whole model structure and its components, including dilated convolution, squeeze-and-excitation block, feature fusion, and dimensionality reduction head.

In next two chapters, we introduce two datasets and conduct experiments to show the advantages of our proposed method.

Chapter 5

Datasets and Experimental Settings

In this chapter, we introduce two datasets: the modified CIFAR-10 dataset and the pediatric polymicrogyria MRI (PPMR) dataset, clarify evaluation metrics, and demonstrate the experimental setting for these two datasets.

5.1 Dataset Description

We provide a modified CIFAR-10 dataset setting to mimic the main challenges of our research; specifically minority class samples in the training data are not representative. As training the PPMR dataset is time-consuming, we use a modified CIFAR-10 dataset to perform exhaustive experiments, comparisons, and analysis; and use the best loss function on the PPMR dataset. The modified CIFAR-10 dataset settings and codes are open-sourced, so everyone can reproduce these experiments or modify the code for their own

usage. The anonymized PPMR dataset will also be made available. More details of these two datasets are given in the following sections.

5.1.1 Modified CIFAR-10 Dataset

CIFAR-10 dataset is a collection of color images with the resolution of 32×32 grouped into ten classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. Each class contains 5,000 training images and 1,000 testing images. To mimic binary classification or anomaly detection, we pick one class as normal (negative) samples, and the rest of the classes as anomalous (positive) samples. In addition, we split these 5,000 training images into training data and validation data with a ratio of 8:2.

Since our PPMR dataset is imbalanced and lacks data diversity, we modify the CIFAR-10 dataset to make it an imbalanced dataset. The imbalanced ratio in the modified CIFAR-10 dataset should be slightly different from the ratio between normal and anomalous samples in our PPMR dataset, which is 5:1. In our PPMR dataset, the diversity of anomalous samples is small because there are multiple consecutive MRIs which are similar to each other. If we employ the same imbalanced ratio, the diversity of anomalous samples in the modified CIFAR-10 dataset ought to be larger than the anomalous diversity in the PPMR dataset. This is the main reason why we use different imbalanced ratios of these two datasets. Hence, we set the imbalance as 10:1 of normal to anomalous samples. After modification, the amount of normal training data is $5,000 * 80\% = 4,000$ and the amount of anomalous training samples is $4,000 * 10\% = 400$. The amount of normal validation data is $5,000 * 20\% = 1,000$ and the amount of anomaly validation data is $1,000 * 10\% = 100$. As

for testing data, we want to observe the distribution of prediction values on the whole testing data, so we have not removed any data samples from testing data. Hence, the amount of normal testing data is $1,000$; and the amount of anomaly testing data is $1,000 * 9 = 9,000$.

Moreover, we pick one class as normal, and the rest of the 9 classes as anomalies, but only 5 anomaly classes appear in training and validation data. We call these classes that appear in training and validation data as seen anomaly. On the other hand, classes that only appear in testing data are called unseen anomaly.

Class ID and label name matches are shown in Table 5.1. One detailed example where the ship class with class ID 8 is regarded as the normal class is shown in Table 5.2. In this case, the normal class has ID 8, seen anomaly classes have IDs 9,0,1,2 or 3 and unseen anomaly class have IDs 4,5,6 or 7.

Class ID	Label Name
0	airplane
1	automobile
2	bird
3	cat
4	deer
5	dog
6	frog
7	horse
8	ship
9	truck

Table 5.1: Class ID and label name matches

Data Splitting	Training	Validation	Testing
Normal Data Amount	4,000	1,000	1,000
Anomaly Data Amount	400	100	9,000
Normal Class ID	8	8	8
Anomaly Class IDs	9,0,1,2,3	9,0,1,2,3	9,0,1,2,3,4,5,6,7

Table 5.2: Data splitting example

5.1.2 Pediatric Polymicrogyria MRI (PPMR) Dataset

The Research Ethics Board of the Children's Hospital of Eastern Ontario gave its approval to the research protocol. Due to the study's retrospective nature, informed consent was waived. We looked for PMG in the hospital's radiology information system. The MRI studies performed on these patients were searched for in the hospital's picture archiving and communication system (PACS).

A pediatric neuroradiologist with 12 years of experience reading children's brain MRIs reviewed the studies. He is a fellow in pediatric radiology. The patient was included in the study if PMG was found to exist. The highest resolution sequence for displaying cortical detail, the coronal 3D gradient echo T1 weighted sequence, was exported as JPEG images from the PACS system. One examination was performed on each patient. The most recent study was picked if there were many studies. Moreover, the imaging parameters for the coronal T1 weighted sequence are shown in Table 5.3.

	3T Skyra magnet	1.5 Tesla Cigna magnet
machine	Siemens	General Electric

	3T Skyra magnet	1.5 Tesla Cigna magnet
repetition time (TR)	2200 ms	10.44 ms
inversion time (TI)	1030 ms	450 ms
echo time (TE)	2.63 ms	4.3 ms
matrix	320*260	512*512
slice thickness	1.2 mm	1.2 mm
FOV	20*23 cm	22*27 cm

Table 5.3: Imaging parameters for the coronal T1 weighted sequence

Three control subjects were chosen for each patient and matched for gender and age (within 6 months). The coronal 3D gradient echo T1 weighted sequence was chosen for the patients. The images anterior to the mid-coronal plane of the orbital globes and posterior to the torcula herophili were discarded because they were insufficient for displaying grey and white matter. The radiologist went over the patient's MRI slices once more and labelled whether PMG was present or not (1 if present, 2 if absent). Since this dataset is labelled only by one senior radiologist but different experts may annotate differently. Especially confusing slices may not always be well-labelled. We can regard these slices as noisy data.

The incidence of polymicrogyria is unclear and rare [114]. Hainc et al. [115] analyzed 819 patients they encountered in total starting from 2008 to 2021. Only 14 patients suffered from polymicrogyria, which is a ratio of 1.7%. Based on a brief analysis of a database from the hospital, Dr. Abdeen estimated the population incidence of polymicrogyria to be 0.01%.

The PPMR dataset contains 23 patients in total. For each patient, we collect 3 age and gender-matched controls because it reduces the skew in the dataset. In each patient's brain, some MRI slices show anomalies and hence are anomaly slices but some other slices

are normal. Furthermore, anomaly slices often occur in the center of the transverse axis of the patient's brain, and normal slices are normally on two sides of the brain. Although the ratio between controls and patients is 3:1, the ratio between normal slices and anomaly slices is around 5:1. The ratio between the amount of normal MRI slices and the amount of anomaly MRI slices for each patient are shown in Figure 5.1. And the whole amount of normal MRI slices and anomaly MRI slices for each patient with 3 controls are shown in Figure 5.2. We can see this is an imbalanced dataset.

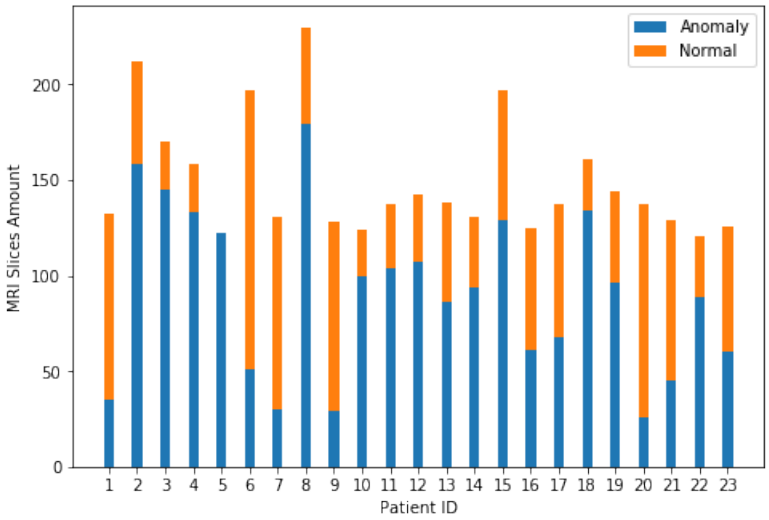


Figure 5.1: The amount of normal slices and anomaly slices for each patient

Each patient's brain includes around 150 scans on average. For example, the 3D view of one pediatric brain is shown in Figure 5.3, and all MRIs of this brain are shown in Figure 5.4. Besides, consecutive PMG scans are similar, so the whole dataset lacks disease diversity. Since we design a specific 5-fold CV for the PPMR dataset, which includes inner and outer folds and is demonstrated in detail in Section 5.3.3, we will also split these 23 patient brains into training, inner fold validation, and outer fold validation data,

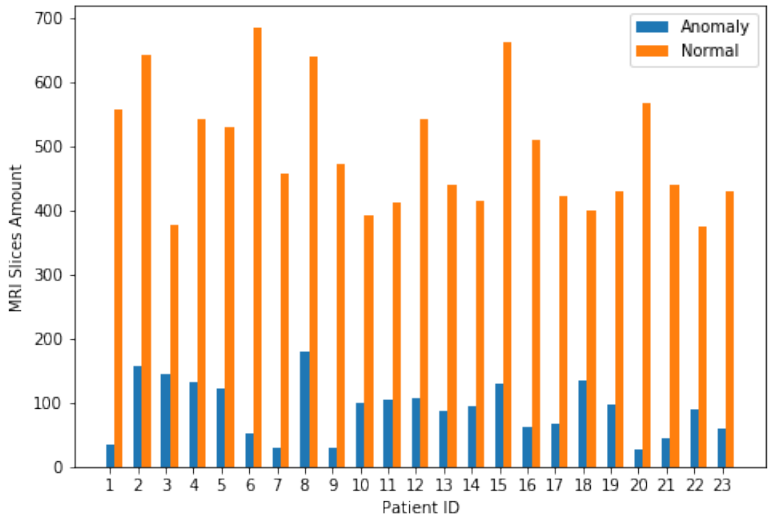


Figure 5.2: Our hospital polymicrogyria dataset MRI slices distribution

with a ratio of 15:4:4. Overall, PMG data are limited, and may not represent the whole distribution of anomaly samples. Even worse, features of PMG scans in inner fold validation or outer fold validation data could be different from features of PMG scans in training data. We regard PMG scans in outer fold validation data which have similar features to PMG scans in training data as 'seen anomaly'. On the other side, PMG scans in outer fold validation data which have dissimilar features are called 'unseen anomaly'. This is a small dataset for deep learning.

To avoid data leakage, we divide the whole dataset into training, inner fold validation, and outer fold validation data at the patient level, because this kind of data split method is more appropriate, compared with the slice-based split [116, 117]. In other words, all PMG scans of one certain patient must be only in training, inner fold validation, or outer fold validation data exclusively. The patient-based data splitting method makes working

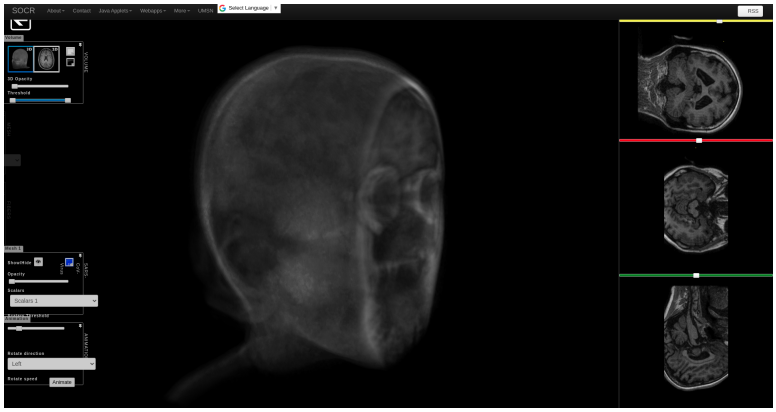


Figure 5.3: 3D view of one pediatric brain

with the small dataset more challenging but is crucial because of the similarity between neighboring slices for a patient.

5.2 Metric and Evaluation

Accuracy, precision, recall, F_β measure, area under the curve of ROC are commonly used to evaluate the performance of binary classification tasks. These formulas are listed in Eqn. 5.5 to Eqn. 5.8.

$$TP : TruePositive \tag{5.1}$$

$$TN : TrueNegative \tag{5.2}$$

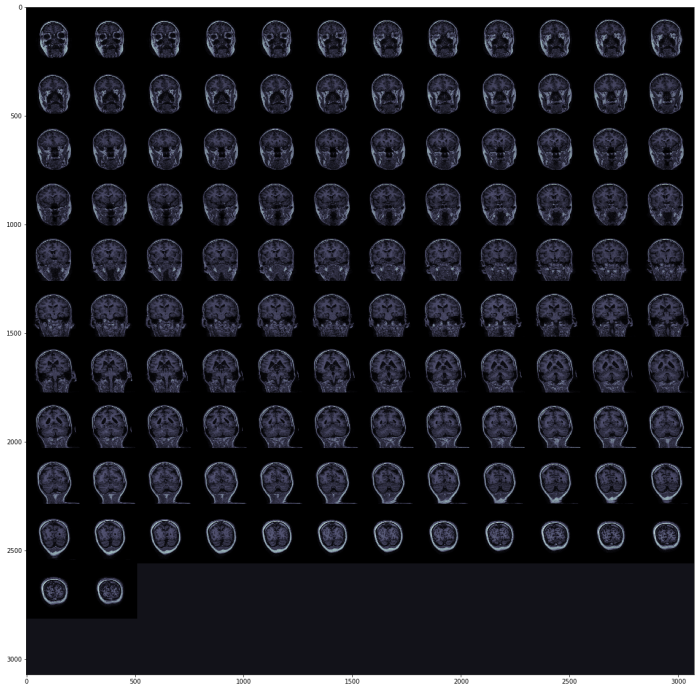


Figure 5.4: All MRIs of one pediatric brain

$$FP : FalsePositive \tag{5.3}$$

$$FN : FalseNegative \tag{5.4}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.5}$$

$$Precision = \frac{TP}{TP + FP} \tag{5.6}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.7}$$

$$F_\beta = (1 + \beta^2) * \frac{Precision * Recall}{\beta^2 * Precision + Recall} \tag{5.8}$$

Specifically, β is a factor which controls the importance between precision and recall. When $\beta = 1$, F_1 measure is the harmonic mean of precision and recall. Since we consider recall more important than precision in our task, we also consider $\beta > 1$. For example, when $\beta = 2$, F_2 measure means recall is twice as important as precision. Two formulas are listed in Eqn. 5.9 and Eqn. 5.10.

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5.9}$$

$$F_2 = \frac{5 * Precision * Recall}{4 * Precision + Recall} \tag{5.10}$$

The recall, also known as sensitivity, or true positive rate, is regarded as the most important metric in medical image classification [118]. The purpose of this research project is to assist radiologists with detecting PMG from children's magnetic resonance imaging (MRI) brain scans. The goal for our model is to miss as few positive (anomaly) samples as possible. It is unreasonable to only take recall into account because it is trivial for a

classifier to attain 100% recall by simply predicting all samples as positive. In our opinion, we consider recall more important than precision, so the F_2 measure is chosen as the crucial evaluation metric. In addition, we will also report other metrics such as recall, precision, F_1 measure, accuracy, area under receiver operating characteristic (ROC) curve [119], etc. If there exists one deep learning model which has a good F_2 measure, meaning a relatively high recall and an acceptable precision, this deep learning model can predict all MRI scans and select some potential PMG disease scans, then, radiologists or doctors can focus more on verifying these scans predicted as PMG by the deep learning model.

5.3 Experimental Settings

5.3.1 Training Details on Modified CIFAR-10 Dataset

Our method uses an end-to-end training strategy. In our novel cDCM loss training, we employ the same LeNet-type [120] convolutional neural networks (CNN) to train our model on the modified CIFAR-10 dataset as Ruff et al. have used in their corresponding Deep SAD experiments. LeNet-type CNN architecture is illustrated in Figure 5.5. The model structure is sequential, which includes 3 CNN blocks and a fully-connected dense layer to map features into a 128-dimensional vector. Each CNN block contains a convolution layer with a kernel of 5×5, a batch normalization layer, a leaky ReLU activation with a leaky ratio of 0.01, and a max-pooling layer with a pooling size of 2×2. We choose the output channel dimensions as 32, 64, 128 respectively in these 3 CNN blocks. More hyper-parameters settings are shown in Table 5.4.

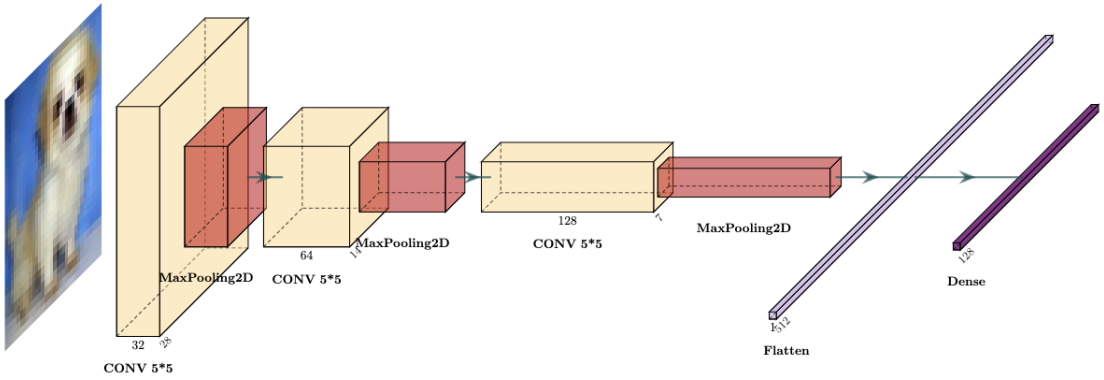


Figure 5.5: LeNet-type CNN

Hyper-parameters	Value
margin	5
theta	5
Latent Representation Dimension	128
CNN Kernel Size	5
Leaky ReLU ratio	0.01
Optimizer	Adam
Initial Learning Rate	1e-3
Batch Size	128
Maximum Training Epochs	400

Table 5.4: Hyper-parameters setting for the modified CIFAR-10 dataset

As for the cross-entropy-based loss training, based on the structure in Figure 5.5, we add one more fully-connected dense layer to transform the 128-dimensional vector into one probability logit value with the help of a sigmoid activation.

Since modified CIFAR-10 training and validation data lack diversity, training models is easy for a model to overfit to the training data. To relieve overfitting, we choose four different data augmentation methods (i.e., rotate, shift, shear, and zoom) during training.

In addition, since our PPMR dataset is small, training deep learning models on this dataset is easy to lead to overfitting as well. We minimize the loss, however when evaluating the corresponding model parameters on the validation set, we rely on the AUCROC metric. Therefore, we save the model parameters with the best metric on the validation data during training and keep this model eventually for inference. After inference, we find we can get a high recall and acceptable precision. There are several reasons why we choose the best validation AUCROC rather than other validation metrics during training:

- If we choose the best validation recall during training, the model tends to predict all samples as anomalies, which is 100% recall, so this trained model is meaningless.
- We do not want to choose the best validation precision during training, mainly because although precision is necessary, we focus more on recall in this research.
- Accuracy is not meaningful in our task because the modified CIFAR-10 dataset and our PPMR dataset are imbalanced.
- The best F_1 measure does not always represent good recall and acceptable precision. A good F_1 measure may represent good precision but not high recall, or it may represent a medium precision and recall, neither is a good solution for our application.
- Although the F_2 measure can represent a high recall and an acceptable precision, the distributions between validation data and testing data are shifted due to unseen anomalies. Hence, an optimal threshold to achieve a high F_2 measure from validation data may not be a good threshold for the testing data. This will be investigated in more details in Chapter 6.

During training, we decay the learning rate with a ratio of 0.5 when validation loss is on a plateau (no decrease) within 900 iterations. Moreover, since the number of training epochs is one important hyper-parameter setting, we set a relatively large number of training epochs, i.e., 400, and apply early stopping when the validation loss does not decrease within 4,500 iterations. Early stopping helps to save a lot of computational resources and time during training.

Since we use the same testing data for all models in the CIFAR-10 dataset, it is not necessary to do cross-validation in our modified CIFAR-10 dataset. To obtain more stable results, we train each model 5 times with random weight initialization and then average the results of these 5 experiments. Models on the modified CIFAR-10 dataset are implemented in TensorFlow 2.8 and trained on Google Colab Pro+.

5.3.2 Training Details on Our Pediatric Polymicrogyria MRI (PPMR) Dataset

In this section, we illustrate training details on our PPMR dataset. The input image shape is $256 \times 256 \times 3$, and these images are normalized to the range from 0 to 1. More hyper-parameters settings are given in Table 5.5.

Hyper-parameters	Value
margin	5
theta	5
center	All 1.0
Latent Representation Dimension	128
CNN Kernel Size, except Dilated CNN	3

Hyper-parameters	Value
Leaky ReLU ratio	0.01
Dropout ratio	0.2
Optimizer	Adam
Initial Learning Rate	1e-3
Batch Size	64
Maximum Training Epochs	200

Table 5.5: Hyper-parameters setting for the PPMR dataset

Some work [121, 122] used skull stripping as one pre-processing step on the brain tumor challenges. But we do not choose this pre-processing step because it may lose some essential information of PMG, such as parts of gray matters connected with the brain skull, which is shown in Figure 5.6. In addition, skull stripping could cause some errors because it removes some important areas. One example is shown in Figure 5.7, and we can see almost half of the brain area is removed. On average, around a quarter of MRIs after skull stripping are failure cases with the PyPI DeepBrain package, and the majority of these failure cases often occur on two sides of the transverse axis of the patient's brain. While other packages may have higher success rates, we chose not risking to invalidate any data of our small dataset.

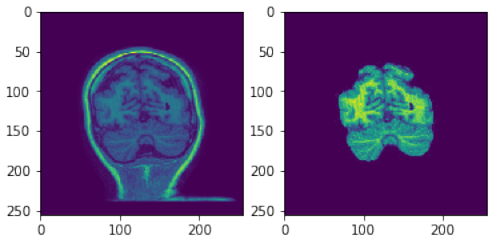


Figure 5.6: Relatively good skull stripping. Left is the original image and right is the image after skull stripping. This skull stripping uses PyPI DeepBrain package.

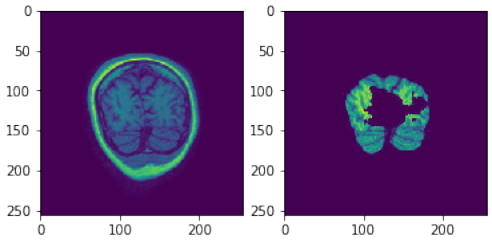


Figure 5.7: Bad skull stripping. Left is the original image and right is the image after skull stripping. This skull stripping uses PyPI DeepBrain package.

We trained our customized deep learning model only on our PPMR dataset, mainly because the LeNet-type model is too simple and not suitable for the complex polymicrogyria classification task, but our customized deep learning model is too complex for CIFAR-10 dataset classification tasks. Hence, we trained two different models for two different datasets. Models on our PPMR dataset are implemented in TensorFlow 2.4 version and trained on a single NVIDIA GeForce RTX 3090 24GB GPU. We use the batch size of 64 in PPMR dataset rather than 128 in the modified CIFAR-10 dataset to fit the model and data into memory. In addition, training strategies, including data augmentation, learning rate decay on a plateau, early stopping, saving model weights with the best validation AUCROC, and so on, are the same as for the modified CIFAR-10 dataset training setting.

Different from averaging five randomly initialized training results on the modified CIFAR-10 dataset, we choose a specific cross-validation to record the performance results on PPMR dataset.

5.3.3 Cross Validation (CV)

We utilize a specific 5-fold cross-validation in our PPMR dataset. Our specific 5-fold cross-validation includes the inner and outer fold inspired by the double cross-validation. Different from double cross-validation, we utilize four inner folds with different validation data for each outer fold. Since our whole PPMR dataset is relatively small, validation data may be biased or be not representative. In addition, the poor quality of validation data will affect model generalization. If we want to obtain a model with good generalization, we have to save model weights where validation performance is at least good enough. Therefore, in each outer fold inference phase, we pick the model with the best validation AUCROC value in these 4 inner folds.

5.4 Summary

In this chapter, we introduce two datasets: one is our PPMR dataset and another one is the modified CIFAR-10 dataset. The modified CIFAR-10 dataset is imbalanced and the classes are only partly seen in the training data by the model, to mimic the challenge of our PPRM dataset. We also introduce evaluation metrics in this chapter. In addition, we illustrate training configuration and strategy details based on these two datasets. In the next chapter, we present, analyze, and discuss experimental results.

Chapter 6

Experimental Results, Analysis and Discussion

6.1 Overview

In this chapter, we show comprehensive experimental results. To reveal the advantages of our novel cDCM loss function, we perform experiments to compare our novel cDCM loss function with cross-entropy-based loss functions on the modified CIFAR-10 dataset in Section 6.2. Because our novel cDCM loss function is inspired by the Deep SAD loss function, we also perform experiments to compare our novel cDCM loss function with the Deep SAD loss function on the modified CIFAR-10 dataset in Section 6.3. We show that our novel cDCM loss function surpasses the state of the art on our modified CIFAR-10 dataset. In addition, we conduct the tests of statistical significance between 5 different loss functions

on the modified CIFAR-10 dataset in Section 6.4. Moreover, we also perform experiments on the modified CIFAR-10 dataset in Section 6.5 to verify that choosing different centers in our novel cDCM loss function does not affect the final performance too much. Additionally, choosing different margins in our novel cDCM loss function also does not affect the final performance too much and these results are given in Section 6.6. As we show the advantages of our novel cDCM loss function in previous experiments, we then apply our novel cDCM loss function to the PPMR dataset and apply our customized deep learning model to the PPMR dataset. In addition, in Section 6.7, we also perform experiments to compare our custom deep learning model with the two popular CNN models EfficientNet and ResNet50 in order to demonstrate the usability of our custom model structure. In Section 6.8, we conduct an ablation study on our custom deep learning model structure to verify the importance of each component. Finally, we carry out some results analysis of the current best method on the PPMR Dataset in Section 6.9.

6.2 Comparison with Cross-Entropy-based Loss Functions

In this section, we compare the modified CIFAR-10 dataset performance of our novel cDCM loss function and several cross-entropy-based loss functions, i.e., binary cross-entropy loss (BCE) in Eqn. 6.1, weighted binary cross-entropy loss (WBCE) in Eqn. 6.2, and focal loss [123] in Eqn. 6.3.

The definitions of the loss functions are as follows:

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}(y_i * \log \hat{y}_i + (1 - y_i) * \log(1 - \hat{y}_i)) \ , \tag{6.1}$$

where y is the ground truth and $\hat{y}$ is the prediction; and,

$$L_{WBCE} = -\frac{1}{N}\sum_{i=1}^{N}(\alpha * y_i * \log \hat{y}_i + (1 - \alpha) * (1 - y_i) * \log(1 - \hat{y}_i)) \ , \tag{6.2}$$

where α is the weight to balance the ratio of positive and negative; and,

$$L_{Focal} = -\frac{1}{N}\sum_{i=1}^{N}(y_i * (1 - \hat{y}_i)^{\gamma} * \log \hat{y}_i + (1 - y_i) * \hat{y_i}^{\gamma} * \log(1 - \hat{y}_i)) \ , \tag{6.3}$$

where γ ($\gamma > 0$) is the scaling factor to force deep learning models to learn more on hard samples.

The binary cross entropy loss is commonly used for binary classification tasks, but it is not suitable for imbalanced datasets because this loss function tends to predict samples as the majority class. Weighted binary cross entropy loss puts more weight on the minority class in the loss function, so this loss can force deep learning models to learn more about the minority class. Focal loss focuses on learning hard samples, including misclassified samples and minority class samples. Intuitively, the scaling factor γ can quickly concentrate the model on hard examples, which are misclassified by the model, while automatically de-weighting the contribution of easy examples, which are correctly classified by the model, during training. [123]

The distribution of the testing data and the distribution of the training data are not the same, because training and validation data are not representative in the modified CIFAR-10 dataset. Consequently, we choose the default threshold 0.5 for BCE, WBCE, and focal losses; and choose the hyper-parameter margin as the decision threshold for our novel cDCM loss. Comparison results between our novel cDCM loss function with cross-entropy-based loss function are recorded in Table 6.1. We can see AUCROC values of these 4 loss functions are similar, but our novel cDCM loss function has an advantage at the predefined decision thresholds. In Figure 6.1, we can see our novel cDCM loss achieves better F_2 measure values in all classes, compared with other loss functions. Moreover, our novel cDCM loss function tends to provide a higher recall without reducing precision too much. Moreover, the F_2 measure of the three cross-entropy-based loss functions are relatively low when the bird is the normal class, compared with other normal classes. This is probably because seen anomaly classes are all animals which include cat, deer, dog, frog and horse, but unseen anomaly classes are all vehicles which include ship, truck, airplane and automobile. Seen animal anomaly classes are more similar to the normal class compared with unseen vehicle anomaly classes. Trained models which are optimized by cross-entropy-based loss functions will regard the majority of unseen vehicles as the majority class during inference, which should be the bird. However, our cDCM loss function can alleviate this issue.

Normal Class	Loss	Precision	Recall	F_1 measure	F_2 measure	Accuracy	AUCROC
airplane	Ours	0.9841 $\pm$0.0048	**0.7239** $\pm$0.0594	**0.8328** $\pm$0.0389	**0.7637** $\pm$0.0530	0.7407 $\pm$0.0496	0.8929 $\pm$0.0058

airplane	BCE	0.9929 ±0.0010	0.5423 ±0.0297	0.7010 ±0.0245	0.5962 ±0.0286	0.5845 ±0.0263	0.8870 ±0.0034
airplane	WBCE	0.9919 ±0.0031	0.6127 ±0.0831	0.7547 ±0.0619	0.6622 ±0.0772	0.6467 ±0.0725	0.8935 ±0.0100
airplane	Focal	0.9937 ±0.0011	0.4956 ±0.0941	0.6570 ±0.0845	0.5492 ±0.0931	0.5431 ±0.0837	0.8838 ±0.0049
automobile	Ours	0.9934 ±0.0011	**0.6735** ±0.0172	**0.8026** ±0.0120	**0.7198** ±0.0157	0.7021 ±0.0150	0.8927 ±0.0050
automobile	BCE	0.9978 ±0.0003	0.5552 ±0.0262	0.7131 ±0.0217	0.6091 ±0.0253	0.5985 ±0.0234	0.9162 ±0.0054
automobile	WBCE	0.9972 ±0.0012	0.5915 ±0.0388	0.7419 ±0.0295	0.6436 ±0.0364	0.6308 ±0.0342	0.9197 ±0.0037
automobile	Focal	0.9965 ±0.0012	0.5832 ±0.0279	0.7354 ±0.0220	0.6358 ±0.0265	0.6230 ±0.0244	0.9153 ±0.0027
bird	Ours	0.9576 ±0.0091	**0.5641** ±0.0552	**0.7085** ±0.0417	**0.6140** ±0.0518	0.5849 ±0.0436	0.7310 ±0.0190
bird	BCE	0.9827 ±0.0049	0.2139 ±0.0465	0.3494 ±0.0604	0.2531 ±0.0516	0.2890 ±0.0405	0.7363 ±0.0145
bird	WBCE	0.9815 ±0.0022	0.2534 ±0.0204	0.4025 ±0.0257	0.2975 ±0.0225	0.3238 ±0.0179	0.7483 ±0.0068
bird	Focal	0.9805 ±0.0052	0.1896 ±0.0499	0.3153 ±0.0679	0.2255 ±0.0562	0.2671 ±0.0434	0.7257 ±0.0107
cat	Ours	0.9662 ±0.0038	**0.6053** ±0.0479	**0.7433** ±0.0362	**0.6538** ±0.0448	0.6256 ±0.0396	0.7741 ±0.0056
cat	BCE	0.9813 ±0.0057	0.3566 ±0.0765	0.5190 ±0.0852	0.4074 ±0.0811	0.4145 ±0.0659	0.7679 ±0.0096
cat	WBCE	0.9717 ±0.0155	0.4881 ±0.1877	0.6324 ±0.1519	0.5354 ±0.1787	0.5243 ±0.1552	0.7749 ±0.0186
cat	Focal	0.9825 ±0.0031	0.2944 ±0.0427	0.4516 ±0.0495	0.3419 ±0.0459	0.3601 ±0.0370	0.7548 ±0.0072
deer	Ours	0.9728 ±0.0018	**0.6546** ±0.0418	**0.7819** ±0.0302	**0.7002** ±0.0385	0.6727 ±0.0358	0.8261 ±0.0058
deer	BCE	0.9927 ±0.0024	0.3383 ±0.0255	0.5042 ±0.0285	0.3896 ±0.0271	0.4022 ±0.0227	0.8251 ±0.0059
deer	WBCE	0.9873 ±0.0105	0.4209 ±0.1828	0.5729 ±0.1544	0.4692 ±0.1768	0.4725 ±0.1561	0.8267 ±0.0114

deer	Focal	0.9873 ±0.0126	0.4224 ±0.1726	0.5757 ±0.1492	0.4712 ±0.1681	0.4736 ±0.1461	0.8189 ±0.0094
dog	Ours	0.9852 ±0.0033	**0.6231** ±0.0402	**0.7627** ±0.0292	**0.6722** ±0.0372	0.6522 ±0.0337	0.8624 ±0.0043
dog	BCE	0.9936 ±0.0026	0.4293 ±0.0412	0.5985 ±0.0406	0.4839 ±0.0421	0.4838 ±0.0359	0.8387 ±0.0054
dog	WBCE	0.9902 ±0.0051	0.4856 ±0.0734	0.6489 ±0.0626	0.5397 ±0.0715	0.5324 ±0.0629	0.8439 ±0.0058
dog	Focal	0.9885 ±0.0100	0.4815 ±0.1358	0.6385 ±0.1107	0.5331 ±0.1301	0.5273 ±0.1151	0.8389 ±0.0029
frog	Ours	0.9909 ±0.0016	**0.7090** ±0.0142	**0.8265** ±0.0091	**0.7518** ±0.0126	0.7323 ±0.0117	0.9107 ±0.0056
frog	BCE	0.9985 ±0.0011	0.5077 ±0.0357	0.6724 ±0.0308	0.5628 ±0.0349	0.5561 ±0.0316	0.9066 ±0.0067
frog	WBCE	0.9943 ±0.0043	0.6329 ±0.0969	0.7638 ±0.0734	0.6810 ±0.0896	0.6880 ±0.0924	0.9139 ±0.0042
frog	Focal	0.9968 ±0.0020	0.5651 ±0.0608	0.7196 ±0.0505	0.6180 ±0.0587	0.6069 ±0.0537	0.9154 ±0.0052
horse	Ours	0.9847 ±0.0058	**0.6120** ±0.0689	**0.7528** ±0.0526	**0.6613** ±0.0647	0.6420 ±0.0583	0.8510 ±0.0172
horse	BCE	0.9982 ±0.0008	0.3493 ±0.0399	0.5165 ±0.0430	0.4012 ±0.0419	0.4138 ±0.0356	0.8807 ±0.0075
horse	WBCE	0.9972 ±0.0016	0.3883 ±0.0712	0.5560 ±0.0708	0.4414 ±0.0729	0.4484 ±0.0633	0.8852 ±0.0029
horse	Focal	0.9981 ±0.0004	0.3332 ±0.0186	0.4994 ±0.0208	0.3844 ±0.0198	0.3993 ±0.0167	0.8854 ±0.0050
ship	Ours	0.9880 ±0.0028	**0.8281** ±0.0331	**0.9007** ±0.0187	**0.8557** ±0.0280	0.8362 ±0.0279	0.9412 ±0.0062
ship	BCE	0.9954 ±0.0014	0.6166 ±0.0733	0.7593 ±0.0585	0.6665 ±0.0698	0.6523 ±0.0650	0.9373 ±0.0039
ship	WBCE	0.9940 ±0.0009	0.6714 ±0.0179	0.8013 ±0.0125	0.7179 ±0.0163	0.7006 ±0.0155	0.9395 ±0.0023
ship	Focal	0.9938 ±0.0015	0.6347 ±0.0778	0.7723 ±0.0595	0.6831 ±0.0730	0.6675 ±0.0688	0.9339 ±0.0023
truck	Ours	0.9828 ±0.0020	**0.7610** ±0.0298	**0.8575** ±0.0185	**0.7968** ±0.0260	0.7729 ±0.0250	0.8974 ±0.0076

truck	BCE	0.9955 ±0.0009	0.5057 ±0.0494	0.6694 ±0.0442	0.5604 ±0.0489	0.5530 ±0.0439	0.9006 ±0.0073
truck	WBCE	0.9963 ±0.0008	0.4874 ±0.0378	0.6539 ±0.0340	0.5426 ±0.0375	0.5370 ±0.0336	0.9045 ±0.0030
truck	Focal	0.9954 ±0.0025	0.4793 ±0.0764	0.6441 ±0.0683	0.5337 ±0.0756	0.5292 ±0.0673	0.9032 ±0.0065

Table 6.1: Comparison between our novel cDCM loss function and three cross-entropy-based loss functions on the modified CIFAR-10 dataset. Each result value in the table is the average of 5 experiments with different weight initialization.

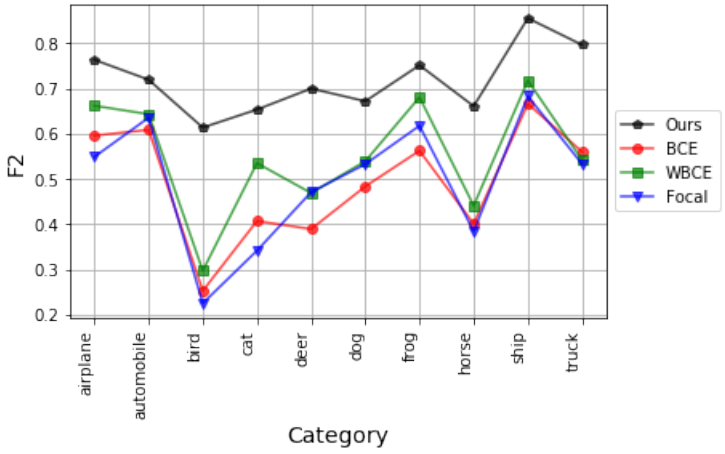


Figure 6.1: Comparison of F_2 measure results between different loss functions on modified CIFAR-10 dataset at the pre-defined decision threholds.

We analyze prediction distributions on modified CIFAR-10 training, validation, and testing data separately. Results are shown in Figure 6.2. Figure 6.2(c) presents the testing data prediction distribution based on the BCE loss. Although more than half of anomaly samples are predicted as anomalous correctly, a large number of anomaly samples are predicted as normal. We can see that this BCE loss model seems to overfit the data; because the predictions on the training data and validation data are perfect (see Figure 6.2(a)

and 6.2(b)), but the prediction on the testing data is poor. Some test samples are predicted incorrectly with high confidence scores. However, using the resulting model obtained with our novel cDCM loss function results in a distribution of testing data predictions in the approximate shape of a normal distribution (see Figure 6.2(f)). Moreover, based on Figure 6.2(c), simply moving the decision threshold for the BCE loss cannot trade-off precision and recall easily because the model trained on the BCE loss fails to recognize too many anomalous samples with high confidence. For example, if one prefers recall to precision, it is reasonable to choose 0.2 or 0.1 as the threshold based on the Figure 6.2(b). However, it does not improve recall too much on testing data.

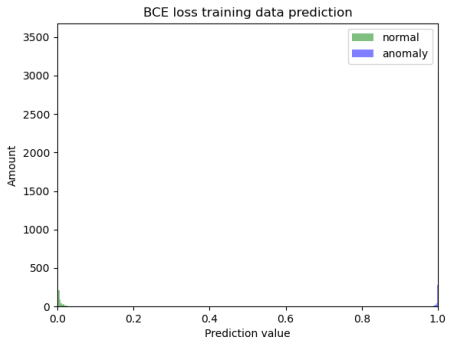
In Figure 6.2(c), we can see prediction values of normal samples are closer to zero than anomaly samples predicted as normal. There is one extreme scenario where all normal samples are predicted as zero, while all anomaly samples are predicted as any values larger than $\frac{1}{+\infty}$ and smaller than or equal to 1. In this scenario, AUCROC value is 1, and this is the main reason why BCE loss can achieve a good AUCROC value.

Figure 6.2(f) represents the testing data prediction distribution based on our novel cDCM loss model. The blue vertical line at a value of 5 on the horizontal axis represents the margin (or decision boundary). We can see only a small part of the samples are predicted incorrectly. Moreover, the majority of normal samples are predicted close to the center c, and the majority of anomaly samples are predicted outside of the decision boundary. In our PPMR dataset, we do not have enough testing data to represent the whole distribution of anomaly samples, and our PPMR testing data are sampled from the whole distribution of anomaly samples. Based on Figure 6.2(f), if testing data are randomly sampled from the whole distribution, recall remains still high, but precision will

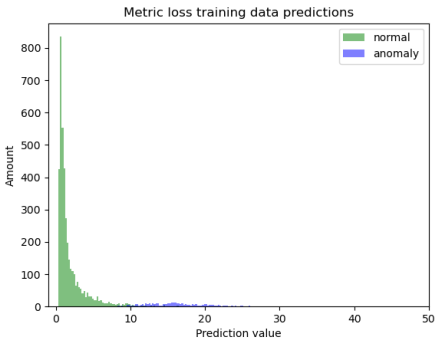
decrease dramatically. This is the main reason why our PPMR dataset tends to result in high recall and high AUCROC, but relatively low precision results when we directly use the predefined margin as the decision threshold. However, precision and recall can be simply balanced by moving the threshold with our novel cDCM loss function, although we prefer recall over precision in our application research.

Moreover, in Fig 6.2(f), the ship is the normal class; truck, airplane, automobile, bird, and cat are seen anomaly classes; and deer, dog, frog, and horse are unseen anomaly classes. In this figure, we can see that the distribution of unseen anomaly samples is on the right side of the distribution of seen anomaly samples. This is mainly because the majority of seen anomaly classes are vehicles, which are more similar to the class ship, compared to animal classes in unseen anomaly classes.

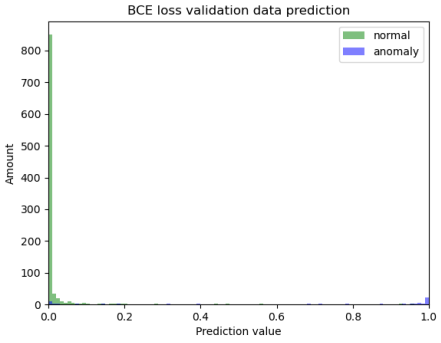
However, in Fig 6.3(c), the airplane is the normal class; automobile, bird, cat, deer and dog are seen anomaly classes; and frog, horse, ship and truck are unseen anomaly classes. In this figure, we can see that the distribution of unseen anomaly samples is on the left side of the distribution of seen anomaly samples, which is different from Fig 6.3(f). This is mainly because the majority of seen anomaly classes are animals, which are dissimilar to the class airplane. In addition, in testing classes, half the classes are animals, but the left half are vehicles.



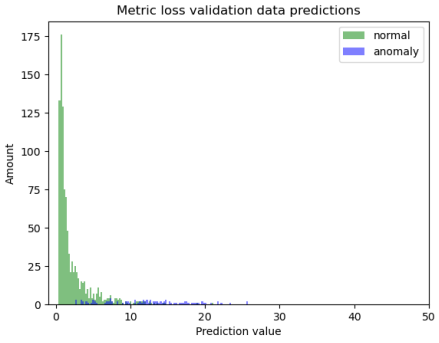
(a) BCE loss training



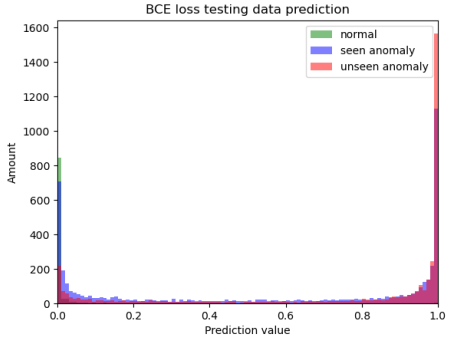
(d) our novel cDCM loss training



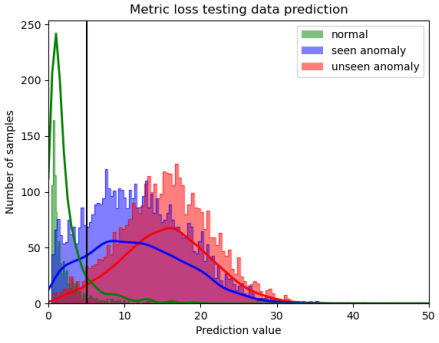
(b) BCE loss validation



(e) our novel cDCM loss validation

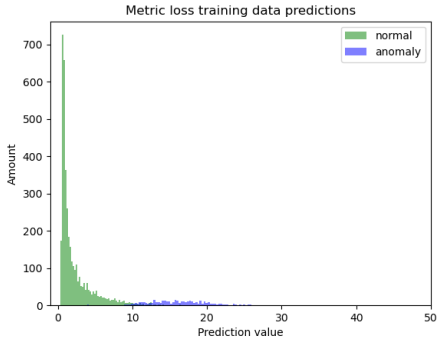


(c) BCE loss testing

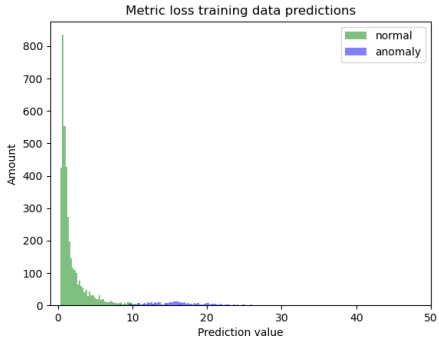


(f) our novel cDCM loss testing

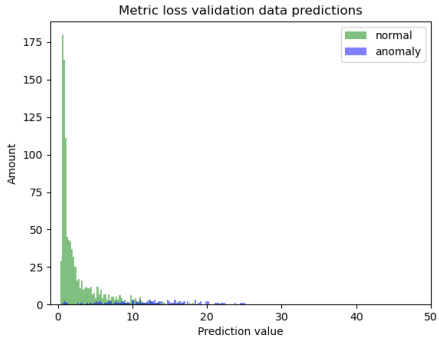
Figure 6.2: Modified CIFAR-10 data samples prediction distribution on BCE loss and our novel cDCM loss. We use the normal class 8 (ship) as a demo. More prediction distribution plots for other normal classes are shown in the Appendix.



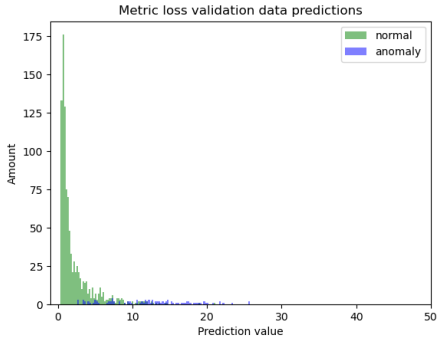
(a) our novel cDCM loss training when the class 0 is normal



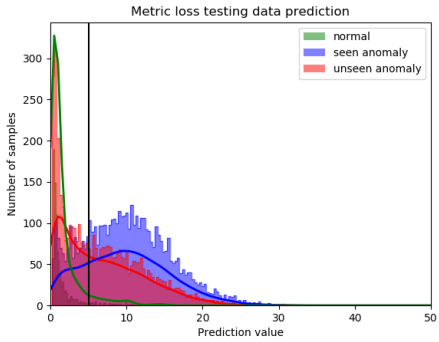
(d) our novel cDCM loss training when the class 8 is normal



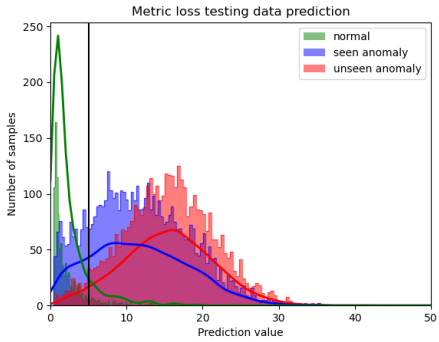
(b) our novel cDCM loss validation when the class 0 is normal



(e) our novel cDCM loss validation when the class 8 is normal



(c) our novel cDCM loss testing when the class 0 is normal



(f) our novel cDCM loss testing when the class 8 is normal

Figure 6.3: Modified CIFAR-10 data samples prediction distribution on our novel cDCM loss. We use the normal class 8 (ship) as one demo in left, and the normal class 0 (airplane) as another demo on right.

6.3 Comparison with Deep SAD

Both the Deep SAD loss and our novel cDCM loss are center-based loss functions. However, the decision threshold for Deep SAD loss is not defined. Zhou et al. [124] use 95% percentile of distance values from all training normal samples to the center as the decision threshold. For this reason, we also choose this method to calculate the decision threshold for Deep SAD.

Comparisons of the modified CIFAR-10 dataset are listed in Table 6.2. We can see recall values, F_1 measure values, F_2 measure values, accuracy values, and AUCROC values of our novel cDCM loss are better than the performance of Deep SAD loss in the majority of classes, although there is a small precision decrease. These results show the advantages of our novel cDCM loss function, compared to the Deep SAD loss function. This is probably because our cDCM loss function has a clear predefined decision threshold which can distinguish confusing samples but the Deep SAD loss function may not disentangle these confusing samples.

Normal Class	Loss	Precision	Recall	F_1 measure	F_2 measure	Accuracy	AUCROC
airplane	Ours	0.9841 ±0.0048	**0.7239** ±0.0594	**0.8328** ±0.0389	**0.7637** ±0.0530	0.7407 ±0.0496	**0.8929** ±0.0058
airplane	Deep SAD	0.9836 ±0.0011	0.7128 ±0.0236	0.8264 ±0.0156	0.7543 ±0.0211	0.7308 ±0.0203	0.8840 ±0.0060
automobile	Ours	0.9934 ±0.0011	0.6735 ±0.0172	0.8026 ±0.0120	0.7198 ±0.0157	0.7021 ±0.0151	**0.8927** ±0.0050
automobile	Deep SAD	0.9876 ±0.0013	**0.7302** ±0.0159	**0.8395** ±0.0103	**0.7703** ±0.0141	0.7489 ±0.0138	0.8744 ±0.0132

bird	Ours	0.9576 ±0.0091	**0.5641** ±0.0552	**0.7085** ±0.0417	**0.6140** ±0.0518	0.5849 ±0.0436	**0.7310** ±0.0190
bird	Deep SAD	0.9667 ±0.0013	0.3246 ±0.0611	0.4838 ±0.0673	0.3739 ±0.0646	0.3823 ±0.0532	0.7197 ±0.0111
cat	Ours	0.9662 ±0.0038	**0.6053** ±0.0479	**0.7433** ±0.0362	**0.6538** ±0.0448	0.6256 ±0.0396	**0.7741** ±0.0056
cat	Deep SAD	0.9750 ±0.0020	0.4769 ±0.1082	0.6340 ±0.1069	0.5290 ±0.1104	0.5181 ±0.0944	0.7609 ±0.0237
deer	Ours	0.9728 ±0.0018	**0.6546** ±0.0418	**0.7819** ±0.0302	**0.7002** ±0.0385	0.6727 ±0.0358	0.8261 ±0.0058
deer	Deep SAD	0.9810 ±0.0028	0.5420 ±0.0599	0.6966 ±0.0496	0.5947 ±0.0578	0.5782 ±0.0515	**0.8272** ±0.0072
dog	Ours	0.9852 ±0.0033	**0.6231** ±0.0402	**0.7627** ±0.0292	**0.6722** ±0.0372	0.6522 ±0.0337	**0.8624** ±0.0043
dog	Deep SAD	0.9849 ±0.0013	0.6172 ±0.0139	0.7587 ±0.0102	0.6670 ±0.0129	0.6469 ±0.0118	0.8454 ±0.0097
frog	Ours	0.9909 ±0.0016	**0.7090** ±0.0142	**0.8265** ±0.0091	**0.7518** ±0.0126	0.7323 ±0.0117	**0.9107** ±0.0056
frog	Deep SAD	0.9904 ±0.0013	0.7047 ±0.0180	0.8234 ±0.0124	0.7478 ±0.0163	0.7281 ±0.0158	0.9027 ±0.0113
horse	Ours	0.9847 ±0.0058	**0.6120** ±0.0689	**0.7528** ±0.0526	**0.6613** ±0.0647	0.6420 ±0.0583	**0.8510** ±0.0172
horse	Deep SAD	0.9844 ±0.0021	0.5916 ±0.0388	0.7384 ±0.0303	0.6427 ±0.0367	0.6240 ±0.0338	0.8236 ±0.0193
ship	Ours	0.9880 ±0.0028	0.8281 ±0.0331	0.9007 ±0.0187	0.8557 ±0.0280	0.8362 ±0.0279	**0.9412** ±0.0062
ship	Deep SAD	0.9835 ±0.0023	**0.8411** ±0.0231	**0.9066** ±0.0130	**0.8661** ±0.0194	0.8443 ±0.0195	0.9265 ±0.0092
truck	Ours	0.9828 ±0.0020	**0.7610** ±0.0298	**0.8575** ±0.0185	**0.7968** ±0.0260	0.7729 ±0.0250	**0.8974** ±0.0076
truck	Deep SAD	0.9824 ±0.0034	0.7521 ±0.0154	0.8518 ±0.0090	0.7891 ±0.0133	0.7648 ±0.0122	0.8769 ±0.0092

Table 6.2: Comparison between our novel cDCM loss function and the Deep SAD loss function on the modified CIFAR-10 dataset. Each result value in the table is the average of 5 experiments with different weight initialization.

Fig 6.4 shows the comparison of F_2 measure results. In the majority of categories, our novel cDCM loss function performs better or slightly better than the Deep SAD loss, except for the category of automobile and ship. In addition, we can see there is a huge performance improvement when the category bird, cat, or deer is the normal class making our model performance more equal between classes. Hence, our novel cDCM loss function is more stable than the Deep SAD loss.

Moreover, Fig 6.5 shows the comparison of AUCROC results. We can see the AUCROC values of our novel cDCM loss function are better than the AUCROC values of the Deep SAD loss function in all categories.

In Table 6.2 and Fig 6.4, we can see when the class 2 (bird) is the normal class, the AUCROC scores of our loss function and Deep SAD loss function are similar (0.731 vs. 0.7197) but the F_2 measure values of these two loss functions have a large difference (0.6140 vs. 0.3739). To analyze this phenomenon, prediction distribution plots on testing data are shown in Fig 6.6. We can see the distribution of Deep SAD loss is narrow, compared to the distribution of our novel cDCM loss when we set their distance distribution on the same scale on the x-axis. Hence, we correct the scale of the maximum value in x-axis from 50 to 5 for the Deep SAD loss and display these new plots in Fig 6.7 for a clear presentation.

In Fig 6.7(c), the black vertical line is the decision threshold, which is the 95% percentile of distance values from all training normal samples to the center. We can see distributions

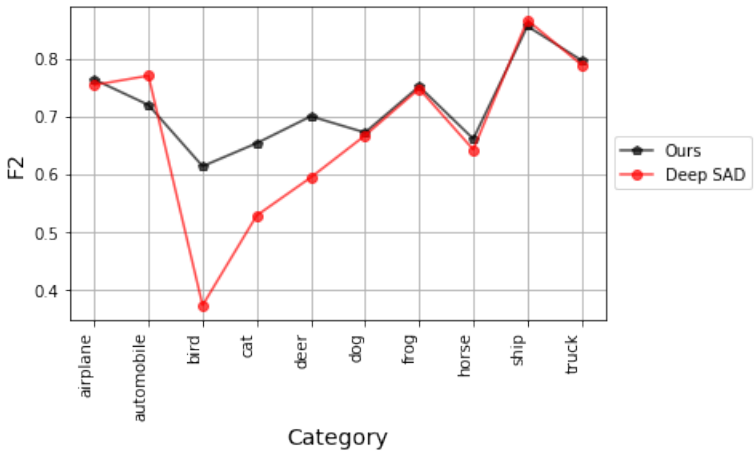


Figure 6.4: Comparison of F_2 measure results between our loss function and Deep SAD loss on modified CIFAR-10 dataset

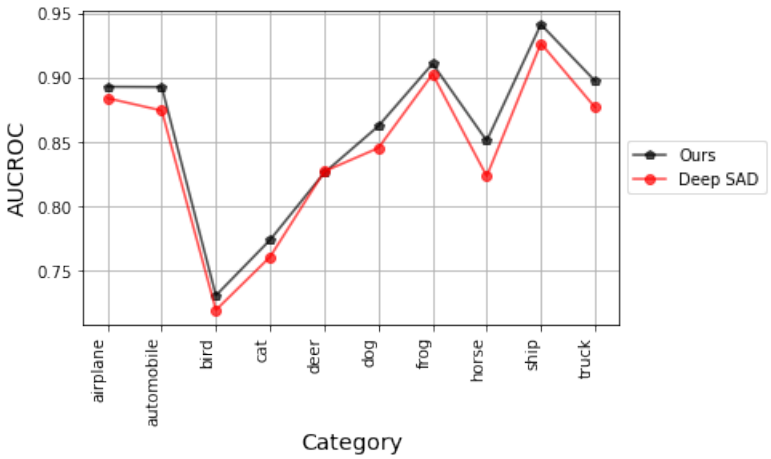
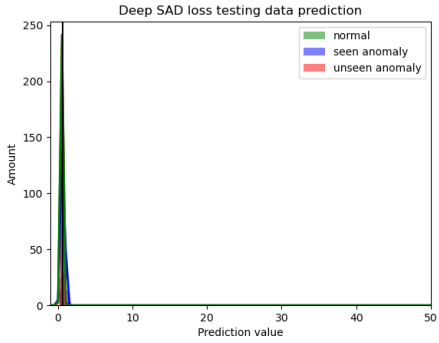
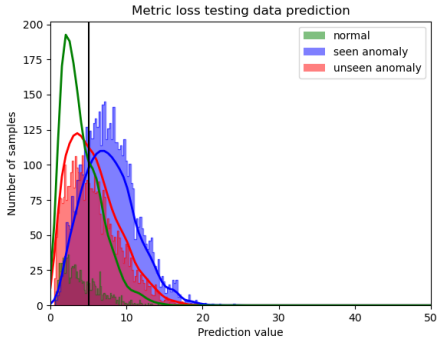


Figure 6.5: Comparison of AUCROC results between our loss function and Deep SAD loss on modified CIFAR-10 dataset



(a) Deep SAD loss testing when the class 2 is normal



(b) our novel cDCM loss testing when the class 2 is normal

Figure 6.6: Modified CIFAR-10 data samples prediction distribution on Deep SAD loss and our novel cDCM loss. The class 2 (bird) is the normal class.

of normal, seen anomaly, and unseen anomaly are close to each other and the mean values of these three distributions are all on the left side of the decision threshold. Moreover, distributions of normal and unseen anomaly are almost the same, which is not beneficial in distinguishing them. Different from the distribution of Deep SAD loss, the distribution of our novel cDCM loss is wide and mean values of seen and unseen anomaly sample distributions are on the right of the decision boundary, while the mean value of normal sample distribution is on the left side of the decision boundary.

Overall, there is probably no good decision threshold for the Deep SAD loss function, but we can define a decision threshold for our novel cDCM loss function first, and the deep learning model will optimize our novel cDCM loss function and separate normal and anomaly distributions into the two sides of the decision threshold. All in all, we develop a method where a good decision threshold can be chosen for imbalanced datasets where the distribution of the anomaly samples is only partially known. This is the main reason why

our novel cDCM loss can achieve better F_2 measure than Deep SAD loss, although they have similar AUCROC scores.

6.4 Tests of Statistical Significance

To verify whether our cDCM loss function has a significant difference from other loss functions on the modified CIFAR-10 dataset, we conduct tests of statistical significance on the F_2 measure metric. We conduct the Friedman Test to verify whether there exists a statistically significant difference between these five loss functions in Section 6.4.1, and then we conduct pair-wise statistical significance tests, which is the Bonferroni-Dunn Test, in Section 6.4.2.

6.4.1 Friedman Test

The Friedman Test [125] can be applied for testing the difference in performance of multiple classifiers over multiple datasets. We have 5 loss functions against 10 datasets. The null hypothesis is that all loss functions perform equally. Table 6.3 shows F_2 measure values and Table 6.4 shows ranks of F_2 measure values of loss functions on each dataset.

Normal Class	Ours	BCE	WBCE	Focal	Deep SAD
airplane	0.7637	0.5962	0.6622	0.5492	0.7543
automobile	0.7198	0.6091	0.6436	0.6358	0.7703
bird	0.6140	0.2531	0.2975	0.2255	0.3739
cat	0.6538	0.4074	0.5354	0.3419	0.5290

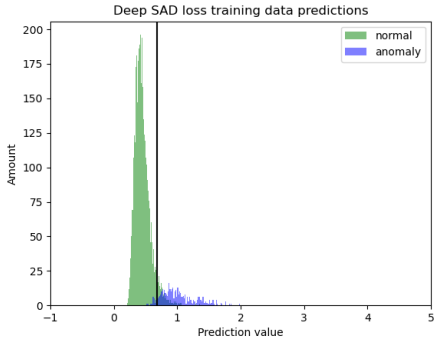
deer	0.7002	0.3896	0.4692	0.4712	0.5947
dog	0.6722	0.4839	0.5397	0.5331	0.6670
frog	0.7518	0.5628	0.6810	0.6180	0.7478
horse	0.6613	0.4012	0.4414	0.3844	0.6427
ship	0.8557	0.6665	0.7179	0.6831	0.8661
truck	0.7968	0.5604	0.5426	0.5337	0.7891

Table 6.3: F_2 measure values from 5 different loss functions on 10 different datasets

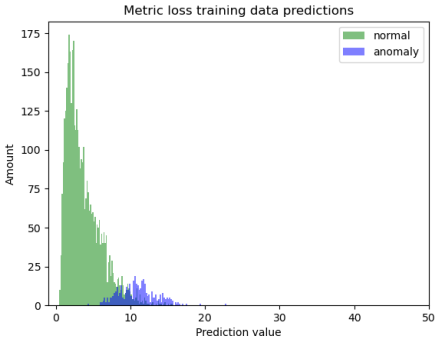
Normal Class	Ours	BCE	WBCE	Focal	Deep SAD
airplane	1	4	3	5	2
automobile	2	5	3	4	1
bird	1	4	3	5	2
cat	1	4	2	5	3
deer	1	5	4	3	2
dog	1	5	3	4	2
frog	1	5	3	4	2
horse	1	4	3	5	2
ship	2	5	3	4	1
truck	1	3	4	5	2
average	1.2	4.4	3.1	4.4	1.9

Table 6.4: Ranks of F_2 measure values from 5 different loss functions on 10 different datasets

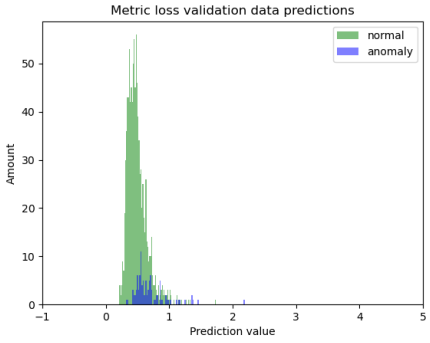
After calculation, the Friedman statistic χ^2 result is 33.52 and the p-value is $9.34e^{-}7$, which is smaller than 0.05. Hence, we reject the null hypothesis and there is a significant difference between the 5 loss functions.



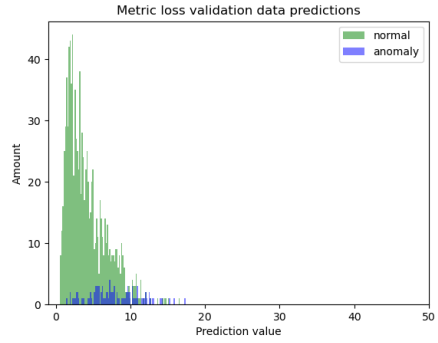
(a) Deep SAD loss training when the class 2 is normal



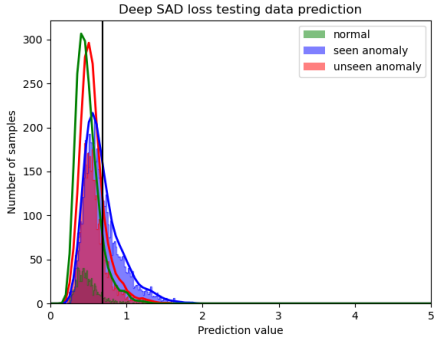
(d) our novel cDCM loss training when the class 2 is normal



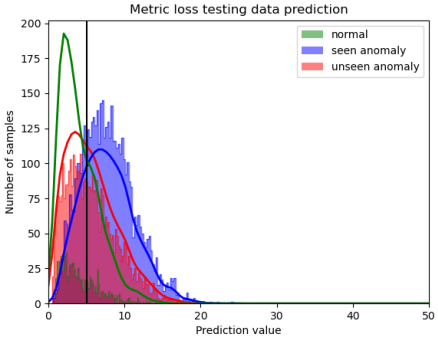
(b) Deep SAD loss validation when the class 2 is normal



(e) our novel cDCM loss validation when the class 2 is normal



(c) Deep SAD loss testing when the class 2 is normal



(f) our novel cDCM loss testing when the class 2 is normal

Figure 6.7: Modified CIFAR-10 data samples prediction distribution on Deep SAD loss and our novel cDCM loss. The class 2 (bird) is the normal class. Please note: the x-axis is range from 0 to 5 in the left three plots, while the x-axis is range from 0 to 50 in the right three plots. The black vertical line is the decision threshold.

6.4.2 Bonferroni-Dunn Test

The Bonferroni-Dunn Test is designed for the significant difference between pair-wise loss functions. After calculation, the critical difference between 5 loss functions is 1.76 at $\alpha = 0.05$. The critical difference diagram for the pair-wise Bonferroni-Dunn Test in shown in Figure 6.8. We can see the cDCM loss function is significantly better than BCE, WBCE and Focal loss functions, but the test shows no significant difference to the Deep SAD loss function at $\alpha = 0.05$.

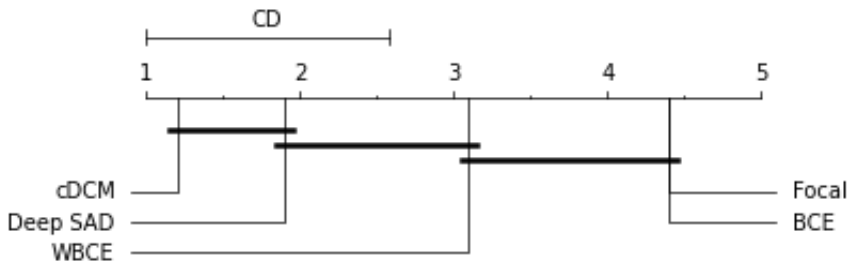


Figure 6.8: Critical difference diagram for the pair-wise Bonferroni-Dunn Test

6.5 Different Centers Comparison

On modified CIFAR-10 dataset, we try three different centers during our novel cDCM loss model training. The first center 128-dimensional all zero vector; the second center is all ones, and the last one is random value drawn from an uniform sampling distribution in the range between 0.0 to 1.0. In Section 4.2, we have shown that the center of our novel cDCM loss can be chosen randomly under the assumption that the model is well trained. In our experiments, we also show that given a specific model structure, choosing different

centers in initialization can achieve almost the same results. In Table 6.5, we can see performance results, especially AUCROC, for different centers are almost the same in all classes. The small difference are likely caused by weight initialization and shuffling batches during training. We save the model with the best validation AUCROC value but a small AUCROC value difference may move the decision threshold by a relatively large amount. This is the likely cause for the performance metrics, except for AUCROC, exhibit small differences in some cases with different centers. However, a slight threshold moving during inference in our novel cDCM loss can increase the precision, whereas a decrease in the recall, or decreasing the precision while increasing the recall. The overall results clearly demonstrate that the center can be chosen randomly in practice.

Normal Class	Center	Precision	Recall	F_1 measure	F_2 measure	Accuracy	AUCROC
airplane	All 0.	0.9837 ±0.0044	0.7264 ±0.0456	0.8349 ±0.0285	0.7661 ±0.0400	0.7427 ±0.0377	0.8896 ±0.0088
airplane	All 1.	0.9841 ±0.0048	0.7239 ±0.0594	0.8328 ±0.0389	0.7637 ±0.0530	0.7407 ±0.0496	0.8929 ±0.0058
airplane	Random	0.9856 ±0.0024	0.7147 ±0.0357	0.8281 ±0.0234	0.7560 ±0.0318	0.7338 ±0.0304	0.8948 ±0.0040
automobile	All 0.	0.9918 ±0.0031	0.6888 ±0.0318	0.8126 ±0.0210	0.7334 ±0.0285	0.7147 ±0.0265	0.8983 ±0.0050
automobile	All 1.	0.9934 ±0.0011	0.6735 ±0.0172	0.8026 ±0.0120	0.7198 ±0.0157	0.7021 ±0.0151	0.8927 ±0.0050
automobile	Random	0.9946 ±0.0015	0.6568 ±0.0397	0.7905 ±0.0284	0.7044 ±0.0364	0.6879 ±0.0347	0.8946 ±0.0050
bird	All 0.	0.9530 ±0.0074	0.6004 ±0.0289	0.7363 ±0.0208	0.6482 ±0.0267	0.6136 ±0.0227	0.7225 ±0.0247
bird	All 1.	0.9576 ±0.0091	0.5641 ±0.0552	0.7085 ±0.0417	0.6140 ±0.0518	0.5849 ±0.0436	0.7310 ±0.0190

bird	Random	0.9548 ±0.0037	0.5754 ±0.0480	0.7170 ±0.0363	0.6246 ±0.0449	0.5932 ±0.0398	0.7261 ±0.0138
cat	All 0.	0.9703 ±0.0043	0.5412 ±0.0495	0.6937 ±0.0391	0.5932 ±0.0470	0.5720 ±0.0414	0.7725 ±0.0126
cat	All 1.	0.9662 ±0.0038	0.6053 ±0.0479	0.7433 ±0.0362	0.6538 ±0.0448	0.6256 ±0.0396	0.7741 ±0.0056
cat	Random	0.9745 ±0.0020	0.5273 ±0.0376	0.6837 ±0.0310	0.5803 ±0.0362	0.5621 ±0.0323	0.7786 ±0.0145
deer	All 0.	0.9711 ±0.0068	0.6908 ±0.0627	0.8058 ±0.0409	0.7324 ±0.0558	0.7029 ±0.0504	0.8378 ±0.0062
deer	All 1.	0.9728 ±0.0018	0.6546 ±0.0418	0.7819 ±0.0302	0.7002 ±0.0385	0.6727 ±0.0358	0.8261 ±0.0058
deer	Random	0.9750 ±0.0033	0.6275 ±0.0374	0.7630 ±0.0270	0.6754 ±0.0345	0.6502 ±0.0311	0.8310 ±0.0074
dog	All 0.	0.9821 ±0.0020	0.6717 ±0.0213	0.7975 ±0.0145	0.7169 ±0.0192	0.6934 ±0.0177	0.8688 ±0.0009
dog	All 1.	0.9852 ±0.0033	0.6231 ±0.0402	0.7627 ±0.0292	0.6722 ±0.0372	0.6522 ±0.0337	0.8624 ±0.0043
dog	Random	0.9840 ±0.0030	0.6415 ±0.0490	0.7757 ±0.0354	0.6891 ±0.0451	0.6678 ±0.0417	0.8645 ±0.0032
frog	All 0.	0.9880 ±0.0067	0.7341 ±0.0384	0.8418 ±0.0236	0.7736 ±0.0335	0.7526 ±0.0307	0.9124 ±0.0186
frog	All 1.	0.9909 ±0.0016	0.7090 ±0.0142	0.8265 ±0.0091	0.7518 ±0.0126	0.7323 ±0.0117	0.9107 ±0.0056
frog	Random	0.9887 ±0.0048	0.7477 ±0.0571	0.8504 ±0.0359	0.7855 ±0.0502	0.7651 ±0.0476	0.9156 ±0.0036
horse	All 0.	0.9863 ±0.0020	0.6030 ±0.0104	0.7484 ±0.0083	0.6538 ±0.0098	0.6352 ±0.0097	0.8604 ±0.0144
horse	All 1.	0.9847 ±0.0058	0.6120 ±0.0689	0.7528 ±0.0526	0.6613 ±0.0647	0.6420 ±0.0583	0.8510 ±0.0172
horse	Random	0.9868 ±0.0036	0.5856 ±0.0567	0.7335 ±0.0441	0.6368 ±0.0536	0.6198 ±0.0485	0.8489 ±0.0176
ship	All 0.	0.9836 ±0.0040	0.8684 ±0.0263	0.9222 ±0.0131	0.8891 ±0.0214	0.8685 ±0.0203	0.9395 ±0.0061

ship	All 1.	0.9880 ±0.0028	0.8281 ±0.0331	0.9007 ±0.0187	0.8557 ±0.0280	0.8362 ±0.0279	0.9412 ±0.0062
ship	Random	0.9859 ±0.0030	0.8618 ±0.0303	0.9194 ±0.0160	0.8839 ±0.0250	0.8645 ±0.0245	0.9439 ±0.0042
truck	All 0.	0.9865 ±0.0015	0.7231 ±0.0214	0.8343 ±0.0143	0.7638 ±0.0191	0.7419 ±0.0193	0.9019 ±0.0085
truck	All 1.	0.9828 ±0.0020	0.7610 ±0.0298	0.8575 ±0.0185	0.7968 ±0.0260	0.7729 ±0.0250	0.8974 ±0.0076
truck	Random	0.9830 ±0.0032	0.7783 ±0.0325	0.8684 ±0.0193	0.8119 ±0.0281	0.7882 ±0.0266	0.9063 ±0.0063

Table 6.5: Comparison between different centers of our novel cDCM loss function on the modified CIFAR-10 dataset. Each result value in the table is the average of 5 experiments with different weight initialization.

6.6 Different Margins Comparison

Margin is another important hyper-parameter in our novel cDCM loss function. On the modified CIFAR-10 dataset, we try two different margins during our novel cDCM loss model training, which are 5 and 10. In Table 6.6, AUCROC values in these two margins are almost same. In addition, we can see in some normal classes, like automobile, deer, and etc., the F_1 measure and F_2 measure on margin 5 is slightly better than the F_1 measure and F_2 measure on margin 10, while in some other normal classes, like dog, frog, and etc., the F_1 measure and F_2 measure on margin 10 is slightly better than the F_1 measure and F_2 measure on margin 5. There is no clear evidence to show which margin is the best, but choosing different margins would not affect the final results too much. Based on our novel cDCM loss function, setting the margin too small results in a tiny hypersphere, while setting the margin too big results in a huge hypersphere. In the worst scenario, setting

margin to 0 or infinity is meaningless. Hence, hyper-parameter "margin" should be set in a reasonable manner.

Normal Class	Margin	Precision	Recall	F_1 measure	F_2 measure	Accuracy	AUCROC
airplane	5	0.9841 ±0.0048	0.7239 ±0.0594	0.8328 ±0.0389	0.7637 ±0.0530	0.7407 ±0.0496	0.8929 ±0.0058
airplane	10	0.9823 ±0.0030	0.7328 ±0.0438	0.8387 ±0.0289	0.7717 ±0.0391	0.7476 ±0.0374	0.8891 ±0.0093
automobile	5	0.9934 ±0.0011	0.6735 ±0.0172	0.8026 ±0.0120	0.7198 ±0.0157	0.7021 ±0.0151	0.8927 ±0.0050
automobile	10	0.9943 ±0.0017	0.6500 ±0.0197	0.7859 ±0.0139	0.6983 ±0.0180	0.6816 ±0.0167	0.8951 ±0.0078
bird	5	0.9576 ±0.0091	0.5641 ±0.0552	0.7085 ±0.0417	0.6140 ±0.0518	0.5849 ±0.0436	0.7310 ±0.0190
bird	10	0.9549 ±0.0049	0.5776 ±0.0710	0.7177 ±0.0510	0.6263 ±0.0653	0.5951 ±0.0582	0.7323 ±0.0132
cat	5	0.9662 ±0.0038	0.6053 ±0.0479	0.7433 ±0.0362	0.6538 ±0.0448	0.6256 ±0.0396	0.7741 ±0.0056
cat	10	0.9623 ±0.0089	0.6230 ±0.0595	0.7547 ±0.0406	0.6695 ±0.054	0.6382 ±0.0460	0.7730 ±0.0114
deer	5	0.9728 ±0.0018	0.6546 ±0.0418	0.7819 ±0.0302	0.7002 ±0.0385	0.6727 ±0.0358	0.8261 ±0.0058
deer	10	0.9754 ±0.0032	0.6280 ±0.0452	0.7632 ±0.0314	0.6758 ±0.0411	0.6508 ±0.0378	0.8308 ±0.0107
dog	5	0.9852 ±0.0033	0.6231 ±0.0402	0.7627 ±0.0292	0.6722 ±0.0372	0.6522 ±0.0337	0.8624 ±0.0043
dog	10	0.9828 ±0.0004	0.6580 ±0.0118	0.7882 ±0.0085	0.7046 ±0.0108	0.6818 ±0.0105	0.8652 ±0.0051
frog	5	0.9909 ±0.0016	0.7090 ±0.0142	0.8265 ±0.0091	0.7518 ±0.0126	0.7323 ±0.0117	0.9107 ±0.0056
frog	10	0.9867 ±0.0017	0.7602 ±0.0222	0.8586 ±0.0140	0.7967 ±0.0195	0.7750 ±0.1920	0.9123 ±0.0073

horse	5	0.9847 ±0.0058	0.6120 ±0.0689	0.7528 ±0.0526	0.6613 ±0.0647	0.6420 ±0.0583	0.8510 ±0.0172
horse	10	0.9884 ±0.0058	0.5724 ±0.0549	0.7236 ±0.0419	0.6245 ±0.0516	0.6089 ±0.0459	0.8606 ±0.0109
ship	5	0.9880 ±0.0028	0.8281 ±0.0331	0.9007 ±0.0187	0.8557 ±0.0280	0.8362 ±0.0279	0.9412 ±0.0062
ship	10	0.9870 ±0.0055	0.8448 ±0.0460	0.9097 ±0.0239	0.8695 ±0.0378	0.8501 ±0.0364	0.9465 ±0.0026
truck	5	0.9828 ±0.0020	0.7610 ±0.0298	0.8575 ±0.0185	0.7968 ±0.0260	0.7729 ±0.0250	0.8974 ±0.0076
truck	10	0.9785 ±0.0041	0.8023 ±0.0297	0.8813 ±0.0166	0.8321 ±0.0251	0.8061 ±0.0241	0.9030 ±0.0110

Table 6.6: Comparison between different margins of our novel cDCM loss function on the modified CIFAR-10 dataset. We use the center of 'All 1.' here. Each result value in the table is the average of 5 experiments with different weight initialization.

6.7 Main Results on the PPMR Dataset

The above experiments show that our novel cDCM loss function is able to predefine a better decision threshold than binary cross-entropy-based loss functions and the earlier Deep SAD loss on the modified CIFAR-10 dataset. Since our PPRM dataset is new and specific, the optimal model structure for our PPMR dataset has not been investigated so far. Since our PPMR dataset is small and imbalanced, normally, transfer learning is suitable for this case. We have experimented with several CNN models that have weights pretrained on ImageNet, including EfficientNet and ResNet50. A series of transfer learning steps are listed below:

- Freezing layers of one pretrained CNN backbone before the global average pooling

layer.

- Adding two fully-connected layers to reduce latent representations to the final n-dimensional vector.

- Training these two fully-connected layers based on our novel cDCM loss function until convergence. More training strategies and configurations are discussed in Section 5.3 and exposed in the source code.

We find if we use these models with transfer learning from ImageNet dataset, these models will predict all samples as positive (anomaly), probably because this leads to the global minimum loss. It means these models fail to learn anything about the new problem domain. It is probably because domain contexts between the ImageNet dataset and our PPMR dataset have a huge difference. Hence, transfer learning from ImageNet dataset is not applicable to our specific challenge.

With our novel cDCM loss function, we customize a model structure based on some insights from MRIs in our PPMR dataset. Our custom model structure combines dilated convolution, squeeze-and-excitation block and feature fusion. We compare our model structure with some popular CNN backbones, i.e., EfficientNet [5] and ResNet50 [126]. Since our PPMR dataset is relatively small and the model may easily overfit the data, light-weight model structures with a limited number of parameters are more reasonable for comparison.

Results are shown in Table 6.7. We can see all metrics, except for recall and F_2-measure, of our custom model structure are worse than metrics of ResNet50. As for ResNet50, moving the threshold can also improve recall with precision decreasing. However, it is

difficult to find an expected threshold without knowing the testing data in practice. Based on our pre-defined decision boundary, our model structure can directly achieve a high recall and an acceptable precision, which means a good F_2-measure. Moreover, we can see standard deviations of all metrics except for accuracy, of our custom model structure are smaller than for the other two models; hence our custom model structure can achieve a more stable results. More importantly, the final results of ResNet50 are unstable, not only because the standard deviations are relatively large, but also because we find that a NAN loss often occurs during training with our novel cDCM loss function. In these cases, one solution is training the model from scratch again. When we train the ResNet50 backbone based on our novel cDCM loss function, we find that, sometimes, this model predicts anomaly samples at a higher and higher distance to the center until the model predicts at least one normal sample as the anomaly, the square of this extremely large distance value will be regarded as part of the loss value, which is NAN. However, we do not find any NAN loss issues during training our custom model, probably because convolutional layers of our custom model structure are not "deep". There is a conjecture that our cDCM loss is more suitable for the "shallow" models, rather than "deep" models. Moreover, another conjecture is that our cDCM loss could be unstable. In our application research, we prefer results with our custom model structure, mainly because the recall is satisfied, precision is acceptable, and results are stable. More importantly, with our novel cDCM loss function, the F_2 measure of our custom model is better than the two backbone models. There is however plenty of room to explore more advanced model structures based on our custom model structure in the future.

Model Blocks	Precision	Recall	F_1 measure	F_2 measure	Accuracy	AUCROC
EfficientNet + cDCM Loss	0.5260 ±0.3243	0.6599 ±0.3990	0.5795 ±0.3481	0.6236 ±0.3746	0.8894 ±0.0518	**0.9535** ±0.0291
ResNet50 + cDCM Loss	**0.6856** ±0.1852	0.7991 ±0.2693	**0.6954** ±0.1394	0.7477 ±0.2140	0.9 ±0.0199	0.9498 ±0.025
Custom Model + cDCM Loss	0.5504 **±0.1586**	**0.9201** **±0.0679**	0.6724 **±0.1106**	**0.7934** **±0.0417**	0.8460 ±0.0885	0.9470 **±0.0226**

Table 6.7: Model comparison: average of 5-fold CV results with standard deviation. All these three models are trained based on our novel cDCM loss function.

For our custom model structure, we also tested a binary cross-entropy loss function and the Deep SAD loss function. Table 6.8 shows the performance in Fold 1 is poor for the two loss functions, especially the performance on the recall metric. After obtaining two folds of results, we have stopped training more folds because the overall performance cannot be good on average anymore and one main goal is to achieve a good recall.

Loss	Fold	Precision	Recall	F_1 measure	F_2 measure	Accuracy	AUCROC
BCE	0	0.7186	0.7398	0.7290	0.7354	0.9058	0.9420
BCE	1	0.6330	0.1869	0.2886	0.2175	0.8573	0.9120
Deep SAD	0	0.4474	0.9049	0.5987	0.7512	0.7924	0.9198

Deep SAD	1	0.7377	0.4878	0.5872	0.5232	0.8938	0.9190

Table 6.8: Some test results of custom model with BCE and Deep SAD loss.

6.8 Ablation Study

To verify each component in our custom model structure is necessary, we perform an ablation study to investigate dilated convolution, feature fusion, and sequeeze-and-excitation blocks. After adding each component, overall model performance increases.

Our ablation study mainly focuses on the model structure for our PPMR dataset, because our custom model structure is too complex and not suitable for our modified CIFAR-10 dataset. We insert model components one by one to illustrate whether the component increases overall performance. Results are shown in Table 6.9. The basic model structure is dilated convolution neural networks, and then we add feature fusion, and finally we add squeeze-and-excitation block. We can see that the averaged recall, F_1 measure and F_2 measure of 5-fold CV increases gradually, and the standard deviation of recall and precision decreases gradually. Overall, after adding feature fusion and squeeze-and-excitation blocks, the recall, F_1 measure and F_2 measure of our model becomes higher and more stable. F_β measure coordinates the precision and recall. Although precision decreases after adding two new components, the F_2 measure increases gradually showing that the model improves overall. Since our PPMR dataset is imbalanced, accuracy is not very meaningful and hence we consider it not in detail. Moreover, AUCROC of the final combined model is also the best.

Overall, after the ablation study, we demonstrate the importance of each component in our custom model structure.

Model Blocks	Precision	Recall	F_1 measure	F_2 measure	Accuracy	AUCROC
DilatedCNN	0.7336 ±0.2171	0.5612 ±0.3166	0.5545 ±0.2459	0.5513 ±0.2873	0.8815 ±0.0442	0.9425 ±0.0240
DilatedCNN +Feature-Fusion	0.5563 ±0.2013	0.8747 ±0.1325	0.6487 **±0.1023**	0.7560 ±0.0668	0.8308 ±0.1141	0.9347 **±0.0165**
DilatedCNN +Feature-Fusion +SE	0.5504 **±0.1586**	**0.9201 ±0.0679**	**0.6724** ±0.1106	**0.7934 ±0.0417**	0.8460 ±0.0885	**0.9470** ±0.0226

Table 6.9: Model Structure Ablation Study. These models are trained based on our novel cDCM loss function.

6.9 Results Analysis of the Current Best Method on the PPMR Dataset

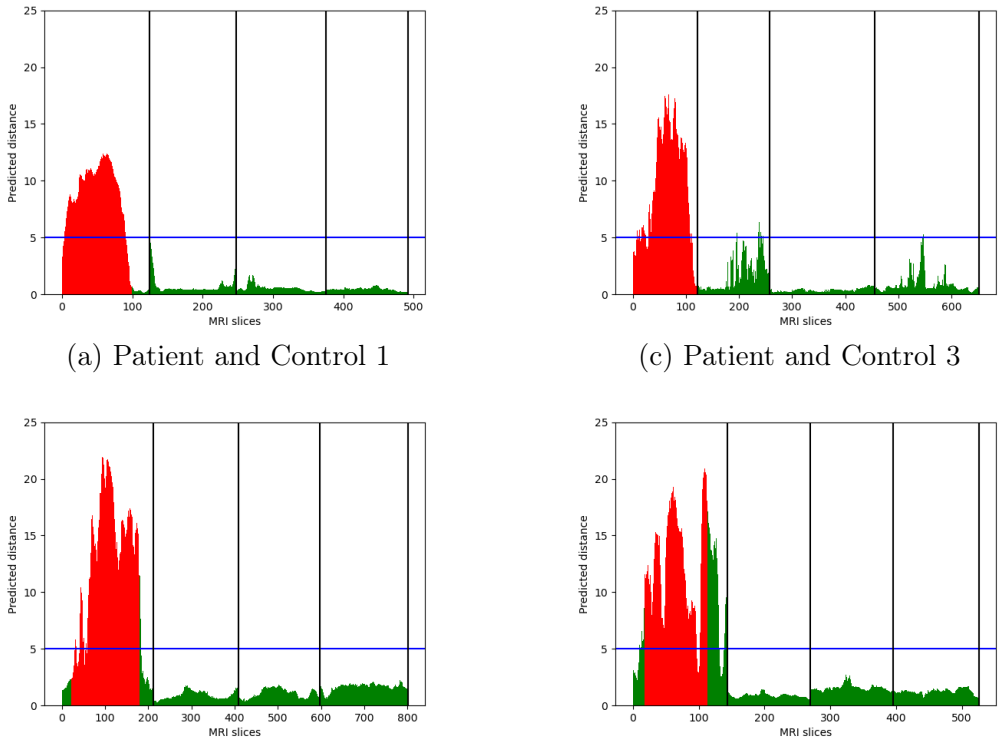
At the moment, our custom deep learning model structure with cDCM loss function achieves the best F_2 measure. Hence, based on this method, we will analyze the prediction distribution of continuous MRI slices for each patient in Section 6.9.1 and analyze the optimal threshold of validation and testing data in Section 6.9.2.

6.9.1 Prediction Distribution Analysis

In this section, we will analyze predicted distance distributions of continuous MRI slices, which include relatively good predictions in Figure 6.9 and relatively poor predictions in Figure 6.10.

In Figure 6.9, we can see distances of the majority of anomaly slices are predicted above the decision threshold, which is 5 in this setting, while the distances of the majority of normal slices are predicted below the decision threshold. In Figure 6.9(c), distances of only a few normal slices are predicted larger than 5. Moreover, in Figure 6.9(d), distances of several continuous normal slices from the brain of one patient are predicted larger than 5, probably because these normal slices are similar to anomaly slices, which confuse the deep learning model. In addition, these normal slices may also be mislabelled partially because these "border" slices are also confused by radiologists.

In Figure 6.10, we can see distances of many normal slices are predicted above the decision threshold. In Figure 6.10(a), distances of almost all normal slices from the patient are predicted above the decision threshold, but all normal slices from controls are predicted correctly. In Figure 6.10(b), distances of a few normal slices are predicted above the decision threshold, while distances of almost half slices from one control brain are predicted above the decision threshold. In Figure 6.10(c), distances of a few border slices from the patient are predicted above the decision threshold. However, distances of several slices, especially the slices in the center of the transverse axis of the patient's brain, are predicted incorrectly above the decision threshold. And these incorrectly predicted slices may occupy the majority of false positive samples. In Figure 6.10(d), distances of several slices on the



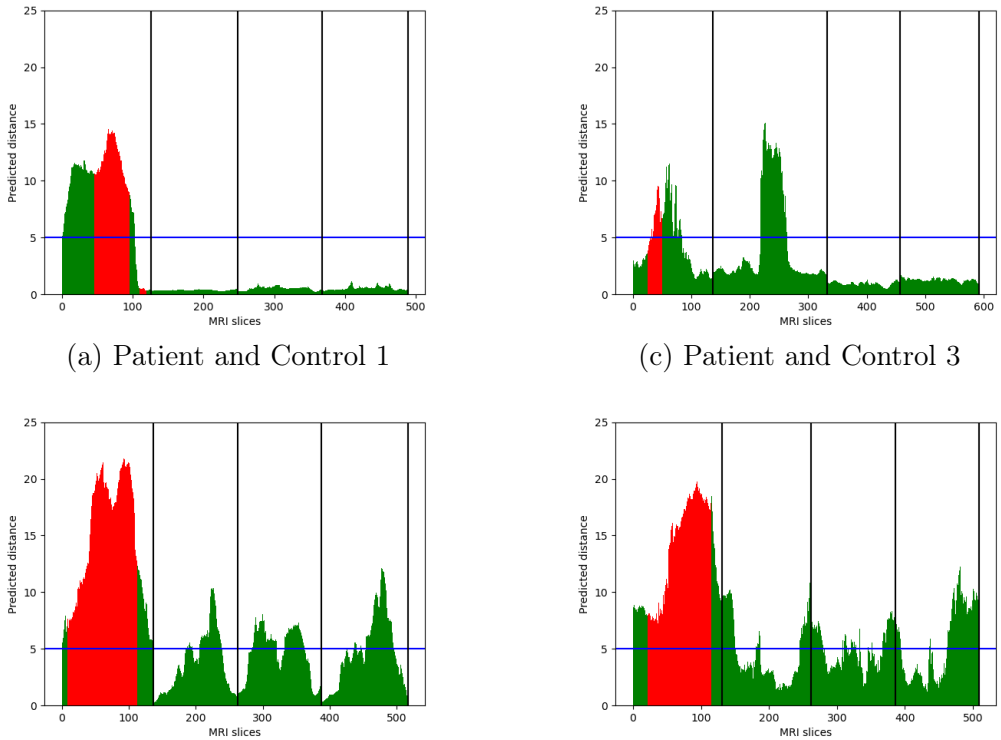
(a) Patient and Control 1

(c) Patient and Control 3

(b) Patient and Control 2

(d) Patient and Control 4

Figure 6.9: Four selected successful classifications in testing data after inference. Each plot shows predicted distances of continuous MRI slices from one patient and three controls. The four brains are separated by vertical black lines. The horizontal blue line represents the decision threshold, which is also the hyper-parameter margin. Normal slices are colored as green, while anomaly slices are colored as red.



(a) Patient and Control 1

(c) Patient and Control 3

(b) Patient and Control 2

(d) Patient and Control 4

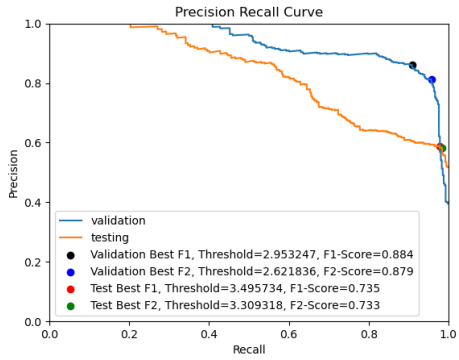
Figure 6.10: Four selected poor classifications in testing data after inference. Each plot shows predicted distances of continuous MRI slices from one patient and three controls. Four brains are separated by vertical black lines. The horizontal blue line represents the decision threshold, which is also the hyper-parameter margin. Normal slices are colored as green, while anomaly slices are colored as red.

two sides of the transverse axis of the patient's brain are predicted above the decision threshold, while distances of slices in the center of the transverse axis of the patient's brain are predicted below the decision threshold. In addition, predicted distance distributions of three controls look like a concave shape, rather than a convex shape. Normally, the predicted distance distribution of MRI slices from patients who suffer from polymicrogyria should have a convex shape because one characteristic of this disease is that it more common in the middle of the brain. Hence, we could regard the prediction who has the concave shape distribution as the control brain, which could be a post-processing step to increase the performance in the future.

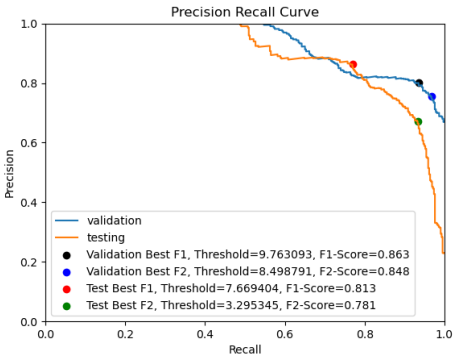
6.9.2 Optimal Threshold Analysis

In this section, we will analyze the optimal decision thresholds from validation data and testing data in different folds. Due to the distribution shift of anomaly samples, the optimal decision thresholds between validation data and testing data may be similar if the shift is small, while the optimal decision thresholds between them may have a huge difference if the shift is too large. Since one property of all normal slices is that these normal slices are similar to each other, the distribution of normal slices is not shifted.

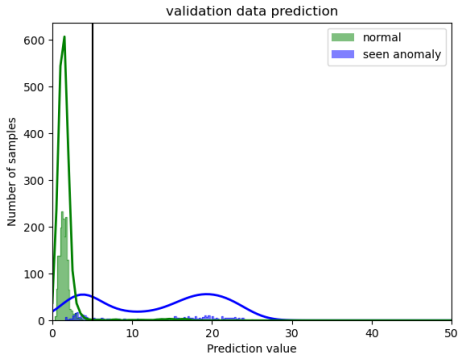
Figure 6.11 shows the results of two folds in 5-fold cross-validation of training customized deep learning model with cDCM loss on the PPMR dataset. For each fold, we present the precision recall (PR) curve on validation and testing data separately, and also show the optimal threshold for each split data on these PR curves. Moreover, we also draw distance prediction distributions on two split data. In Figure 6.11(a), we can see



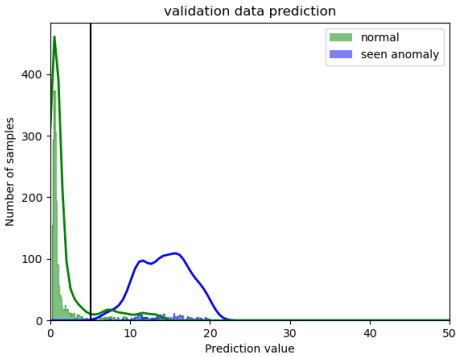
(a) Precision Recall Curve of one fold



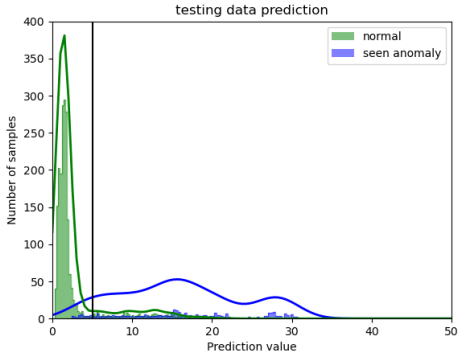
(d) Precision Recall Curve of another fold



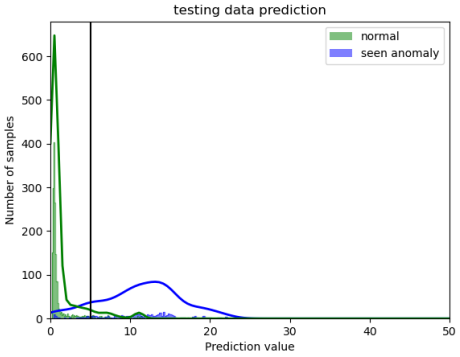
(b) Validation data prediction distribution of one fold



(e) Validation data prediction distribution of another fold



(c) Testing data prediction distribution of one fold



(f) Testing data prediction distribution of another fold

Figure 6.11: Optimal threshold analysis with prediction distribution from validation and testing data on two folds.

optimal thresholds to achieve the best F_1 measure and the best F_2 measure on validation data are 2.9532 and 2.6218 separately; but optimal thresholds on testing data are 3.4957 and 3.3093. There is not a large difference between optimal thresholds from validation and testing data, probably mainly because the change of distance prediction distribution from validation data in Figure 6.11(b) to testing data in Figure 6.11(c) is small. However, in Figure 6.11(d), we can see optimal thresholds between validation and testing data have a relatively large difference, especially, optimal thresholds for the best F_2 measure between the two data changes from 8.4987 to 3.2953. The reason of this phenomenon is probably that the distance prediction distribution of validation data in Figure 6.11(e) is changed to the distribution of testing data in Figure 6.11(f). Hence, choosing the optimal threshold for the validation data as the decision threshold for testing data is not reasonable if the data distribution change happens. As for our default decision threshold, which is 5 in this setting, although it is not better than the optimal threshold of validation data in the first fold because 5 is farther away from 3.3093, compared to the optimal threshold of validation data, which is 2.6218, it is better than the optimal threshold of validation data in another fold because 5 is closer to the optimal threshold of testing data, which is 3.2953, compared to 8.4987. Therefore, our default threshold seems more stable on out-of-distribution samples than the optimal threshold on validation data.

6.10 Summary

In this chapter, we conduct extensive experiments and thoroughly analyze experimental results in detail. On the modified CIFAR-10 dataset, experiments demonstrate the advan-

tages of our novel cDCM loss function. The ability of our novel cDCM loss function to set the decision threshold more appropriately without access to the testing data which is a great benefit. In addition, our experiments show that the two hyper-parameters, center and margin, in our novel cDCM loss function can be assigned randomly in a reasonable manner with few performance changes. Based on our novel cDCM loss function, our experiments show the utility and advantages of our custom deep learning model structure.

Chapter 7

Conclusion and Future Work

Junior radiologists could misclassify polymicrogyria (PMG) MRIs to normal MRIs in practice, mainly because they lack experience but also because of the diversity of PMG. Designing a computer-aided method to detect PMG in MRIs is essential. On the other hand, currently, deep learning models cannot achieve 100% accuracy, so physicians cannot trust deep learning models without any human intervention. In the clinical setting, it is important to identify all cases of PMG, so a high recall is important. A 50% precision can be accepted as there is no clinical penalty, with the only drawback being that the physician must assess the identified images. Our model provides a high recall and an acceptable precision and is therefore promising in the future evaluation of PMG cases.

We find cross-entropy-based loss functions predict the majority of images as normal, which causes results of high precision but low recall. In addition, models with cross-entropy-based loss functions are easy to result in overfitting and they lack the ability to

generalize to testing data, especially, if the testing data contain unseen features during training. Moreover, in our case, we cannot trade-off precision and recall by simply moving the decision threshold if we use cross-entropy-based loss functions. However, our novel cDCM loss function tends to predict images with high F_2 measure, which means relatively high recall and acceptable precision, and partly deals with the generalization problem. Finally, our custom model with our novel cDCM loss function can achieve 92.01% recall and 55.04% precision, where the recall is satisfied and the precision is acceptable in our application research. After comparison, our custom model can achieve better performance than EfficientNet and can get more stable results than ResNet50.

This research has a number of shortcomings. Radiologists will additionally separate out a specific number of normal MRIs from the possible PMG MRIs predicted by our algorithm since the precision of our final results is 55.04%. Furthermore, because of the limited size of our PPMR dataset, it is challenging to assess how well the entire PMG distribution performs in the actual world. However, this is the first time to classify PMG MRIs from normal MRIs by machine learning methods; there is a lot of work that needs to be done in the future.

As for future work, we need more radiology experts to verify the data label to improve the data quality. Our method is promising for some other kinds of medical imaging classification tasks, especially for some tasks whose testing data have out-of-distribution samples. For example, the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset can be tested by our method. Although transfer learning from ImageNet dataset to our PPMR dataset does not work, other medical imaging classification tasks may nevertheless improve the overall performance with the help of transfer learning. Since our novel cDCM

loss is calculated by n-dimensional latent representations, some effective data augmentation methods, like mixup [35], cutmix [36], etc., cannot be applied in our novel cDCM loss method easily. There is a lot of room to explore data augmentation methods based on our novel cDCM loss function.

In our preliminary tests, we generated synthetic images from autoencoders (AE), variational autoencoders (VAE), generative adversarial networks (GAN), and so on. However, these data augmentation methods provide limited help when we train models with cross-entropy-based loss functions in our task, mainly because these data augmentation methods cannot generate unseen features theoretically. These generative models may provide a good regularization during training models with the help of our novel cDCM loss, and they can be explored in the future. Although we already used several regularization methods, such as simple image transformation as data augmentation, dropout, batch normalization, etc., some other regularization methods can be explored for our task in the future. Moreover, precision should be increased further to achieve 70∼80% or higher, without decreasing the recall too much, or keeping the recall above 90%. To achieve this goal, apart from exploring data augmentation, the model structure can also be explored further.

In our cDCM loss function, we applied the L2 norm, which is the Euclidean distance, to calculate the distance from data samples to the center in the vector space. Some other distance calculation formulas can be explored, such as the L1 norm, the cosine similarity, etc. In our preliminary tests, we directly replaced the L2 norm with the L1 norm and cosine similarity, but we found the training failed because of the NAN loss after the replacement. Therefore, the loss function should be customized if different distance formulas are selected in the future.

References

[1] Chenming Li, Zelin Qiu, Xueying Cao, Zhonghao Chen, Hongmin Gao, and Zaijun Hua. Hybrid dilated convolution with multi-scale residual fusion network for hyperspectral image classification. Micromachines, 12(5):545, 2021.

[2] Linrun Qiu, Dongbo Zhang, Yuan Tian, and Najla Al-Nabhan. Deep learning-based algorithm for vehicle detection in intelligent transportation systems. The Journal of Supercomputing, 77(10):11083–11098, 2021.

[3] Alena Selimović, Blaž Meden, Peter Peer, and Aleš Hladnik. Analysis of content-aware image compression with vgg16. In 2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI), pages 1–7. IEEE, 2018.

[4] 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, pages 770–778, 2016.

[5] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning, pages 6105–6114.

PMLR, 2019.

[6] Alaa Sagheer and Mostafa Kotb. Unsupervised pre-training of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems. Scientific Reports, 9(1):1–16, 2019.

[7] A James Barkovich. Current concepts of polymicrogyria. Neuroradiology, 52(6):479–487, 2010.

[8] Louis Maillard and Georgia Ramantani. Epilepsy surgery for polymicrogyria: a challenge to be undertaken. Epileptic Disorders, 20(5):319–338, 2018.

[9] Omneya Attallah, Maha A Sharkas, and Heba Gadelkarim. Deep learning techniques for automatic detection of embryonic neurodevelopmental disorders. Diagnostics, 10(1):27, 2020.

[10] Daichi Sone and Iman Beheshti. Clinical application of machine learning models for brain imaging in epilepsy: a review. Frontiers in Neuroscience, 15:761, 2021.

[11] Erena Siyoum Biratu, Friedhelm Schwenker, Yehualashet Megersa Ayano, and Taye Girma Debelee. A survey of brain tumor segmentation and classification algorithms. Journal of Imaging, 7(9):179, 2021.

[12] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. Advances in Neural Information Processing Systems, 33:18661–18673, 2020.

[13] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122, 2015.

[14] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7132–7141, 2018.

[15] 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, pages 4700–4708, 2017.

[16] Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. arXiv preprint arXiv:1906.02694, 2019.

[17] Ray Hashman Hashemi, William G Bradley, and Christopher J Lisanti. MRI: the basics: The Basics. Lippincott Williams & Wilkins, 2012.

[18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT press, 2016.

[19] Sunitha Basodi, Chunyan Ji, Haiping Zhang, and Yi Pan. Gradient amplification: An efficient way to train deep neural networks. Big Data Mining and Analytics, 3(3):196–207, 2020.

[20] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1):1929–1958, 2014.

[21] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of Statistical Planning and Inference, 90(2):227–244, 2000.

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning, pages 448–456. PMLR, 2015.

[23] Qing Li, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng, and Mei Chen. Medical image classification with convolutional neural network. In 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), pages 844–848. IEEE, 2014.

[24] Wang Zhiqiang and Liu Jun. A review of object detection based on convolutional neural network. In 2017 36th Chinese Control Conference (CCC), pages 11104–11109. IEEE, 2017.

[25] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4):834–848, 2017.

[26] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578, 2016.

[27] Andrew Ng et al. Sparse autoencoder. CS294A Lecture Notes, 72(2011):1–19, 2011.

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

[29] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. arXiv preprint arXiv:1611.01704, 2016.

[30] Lovedeep Gondara. Medical image denoising using convolutional denoising autoencoders. In 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), pages 241–246. IEEE, 2016.

[31] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014.

[32] Hugo Larochelle, Yoshua Bengio, Jérôme Louradour, and Pascal Lamblin. Exploring strategies for training deep neural networks. Journal of Machine Learning Research, 10(1), 2009.

[33] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. arXiv preprint arXiv:2110.11334, 2021.

[34] Suorong Yang, Weikang Xiao, Mengcheng Zhang, Suhan Guo, Jian Zhao, and Furao Shen. Image data augmentation for deep learning: A survey. arXiv preprint arXiv:2204.08610, 2022.

[35] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412, 2017.

[36] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6023–6032, 2019.

[37] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017.

[38] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1):43–76, 2020.

[39] S Deepak and PM Ameer. Brain tumor classification using deep cnn features via transfer learning. Computers in Biology and Medicine, 111:103345, 2019.

[40] Vinay Arora, Eddie Yin-Kwee Ng, Rohan Singh Leekha, Medhavi Darshan, and Arshdeep Singh. Transfer learning-based approach for detecting covid-19 ailment in lung ct scan. Computers in Biology and Medicine, 135:104575, 2021.

[41] Ruhul Amin Hazarika, Ajith Abraham, Samarendra Nath Sur, Arnab Kumar Maji, and Debdatta Kandar. Different techniques for alzheimer's disease classification using brain images: a study. International Journal of Multimedia Information Retrieval, pages 1–20, 2021.

[42] Maximilian E Tschuchnig and Michael Gadermayr. Anomaly detection in medical imaging-a mini review. Data Science–Analytics and Applications, pages 33–38, 2022.

[43] Omneya Attallah, Heba Gadelkarim, and Maha A Sharkas. Detecting and classifying fetal brain abnormalities using machine learning techniques. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pages 1371–1376. IEEE, 2018.

[44] Piotr Płoński, Wojciech Gradkowski, Irene Altarelli, Karla Monzalvo, Muna van Ermingen-Marbach, Marion Grande, Stefan Heim, Artur Marchewka, Piotr Bogorodzki, Franck Ramus, et al. Multi-parameter machine learning approach to the neuroanatomical basis of developmental dyslexia. Human Brain Mapping, 38(2):900–908, 2017.

[45] Yogesh Kumar, Apeksha Koul, Ruchi Singla, and Muhammad Fazal Ijaz. Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda. Journal of Ambient Intelligence and Humanized Computing, pages 1–28, 2022.

[46] Md Manjurul Ahsan, Shahana Akter Luna, and Zahed Siddique. Machine-learning-based disease diagnosis: A comprehensive review. In Healthcare, volume 10, page 541. MDPI, 2022.

[47] Sergio Sanchez-Martinez, Oscar Camara, Gemma Piella, Maja Cikes, Miguel Ángel González-Ballester, Marius Miron, Alfredo Vellido, Emilia Gómez, Alan G Fraser, and Bart Bijnens. Machine learning for clinical decision-making: challenges and opportunities in cardiovascular imaging. Frontiers in Cardiovascular Medicine, 8, 2021.

[48] Marcin Kociołek, Michał Strzelecki, and Rafał Obuchowicz. Does image normalization and intensity resolution impact texture classification? Computerized Medical Imaging and Graphics, 81:101716, 2020.

[49] Pooja Prabhu, A Kotegar Karunakar, Sanjib Sinha, N Mariyappa, GK Bhargava, Jayabal Velmurugan, and H Anitha. Content-based estimation of brain mri tilt in three orthogonal directions. Journal of Digital Imaging, 34(3):760–771, 2021.

[50] Francisco PM Oliveira and Joao Manuel RS Tavares. Medical image registration: a review. Computer Methods in Biomechanics and Biomedical Engineering, 17(2):73–93, 2014.

[51] P Kalavathi and VB Prasath. Methods on skull stripping of mri head scan images—a review. Journal of Digital Imaging, 29(3):365–379, 2016.

[52] Anil K Bharodiya. Feature extraction methods for ct-scan images using image processing. Computed-Tomography (CT) Scan, page 63, 2022.

[53] Richard Bellman. Dynamic programming. Science, 153(3731):34–37, 1966.

[54] Gerard V Trunk. A problem of dimensionality: A simple example. IEEE Transactions on Pattern Analysis and Machine Intelligence, (3):306–307, 1979.

[55] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In International Conference on Database Theory, pages 420–434. Springer, 2001.

[56] Hervé Abdi and Lynne J Williams. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2(4):433–459, 2010.

[57] ER Henry and J Hofrichter. [8] singular value decomposition: Application to analysis of experimental data. In Methods in Enzymology, volume 210, pages 129–192. Elsevier, 1992.

[58] Alan Julian Izenman. Linear discriminant analysis. In Modern Multivariate Statistical Techniques, pages 237–280. Springer, 2013.

[59] William S Noble. What is a support vector machine? Nature Biotechnology, 24(12):1565–1567, 2006.

[60] Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. An introduction to decision tree modeling. Journal of Chemometrics: A Journal of the Chemometrics Society, 18(6):275–285, 2004.

[61] Wei Zhou, Hongqiang Wang, Chunmei Yang, Yang Bai, Dongliang Wang, and Yongfeng Zhan. Decision tree based medical image clustering algorithm in computer-aided diagnoses. Journal of Computational Methods in Sciences and Engineering, 15(4):645–651, 2015.

[62] Chien-Shun Lo and Chuin-Mu Wang. Support vector machine for breast mr image classification. Computers & Mathematics with Applications, 64(5):1153–1162, 2012.

[63] Mohd Fauzi Bin Othman, Noramalina Bt Abdullah, and Nurul Fazrena Bt Kamal. Mri brain classification using support vector machine. In 2011 Fourth International

Conference on Modeling, Simulation and Applied Optimization, pages 1–4. IEEE, 2011.

[64] Taha J Alhindi, Shivam Kalra, Ka Hin Ng, Anika Afrin, and Hamid R Tizhoosh. Comparing lbp, hog and deep features for classification of histopathology images. In 2018 International Joint Conference on Neural Networks (IJCNN), pages 1–7. IEEE, 2018.

[65] Trung Le, Dat Tran, Wanli Ma, and Dharmendra Sharma. A new support vector machine method for medical image classification. In 2010 2nd European Workshop on Visual Information Processing (EUVIP), pages 165–170. IEEE, 2010.

[66] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. Symmetry, 11(9):1066, 2019.

[67] Masayuki Tsuneki. Deep learning models in medical image analysis. Journal of Oral Biosciences, 2022.

[68] Monique F Kilkenny and Kerin M Robinson. Data quality:"garbage in–garbage out", 2018.

[69] Linda Wang, Zhong Qiu Lin, and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. Scientific Reports, 10(1):1–12, 2020.

[70] Samir S Yadav and Shivajirao M Jadhav. Deep convolutional neural network based medical image classification for disease diagnosis. Journal of Big Data, 6(1):1–18, 2019.

[71] Yin Dai, Yifan Gao, and Fayu Liu. Transmed: Transformers advance multi-modal medical image classification. Diagnostics, 11(8):1384, 2021.

[72] Jinseong Jang and Dosik Hwang. M3t: Three-dimensional medical image classifier using multi-plane and multi-slice transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 20718–20729, 2022.

[73] Xiyue Wang, Sen Yang, Jun Zhang, Minghui Wang, Jing Zhang, Junzhou Huang, Wei Yang, and Xiao Han. Transpath: Transformer-based self-supervised learning for histopathological image classification. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 186–195. Springer, 2021.

[74] Yutong Xie, Jianpeng Zhang, Yong Xia, and Qi Wu. Unified 2d and 3d pre-training for medical image classification and segmentation. arXiv preprint arXiv:2112.09356, 2021.

[75] Sergey Korolev, Amir Safiullin, Mikhail Belyaev, and Yulia Dodonova. Residual and plain convolutional neural networks for 3d brain mri classification. In 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), pages 835–838. IEEE, 2017.

[76] Viktor Wegmayr, Sai Aitharaju, and Joachim Buhmann. Classification of brain mri with big data and deep 3d convolutional neural networks. In Medical Imaging 2018: Computer-Aided Diagnosis, volume 10575, pages 406–412. SPIE, 2018.

[77] Alireza Mehrtash, Alireza Sedghi, Mohsen Ghafoorian, Mehdi Taghipour, Clare M Tempany, William M Wells III, Tina Kapur, Parvin Mousavi, Purang Abolmaesumi,

and Andriy Fedorov. Classification of clinical significance of mri prostate findings using 3d convolutional neural networks. In Medical Imaging 2017: Computer-Aided Diagnosis, volume 10134, pages 589–592. SPIE, 2017.

[78] Hayit Greenspan, Bram Van Ginneken, and Ronald M Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. IEEE Transactions on Medical Imaging, 35(5):1153–1159, 2016.

[79] Kitsuchart Pasupa and Wisuwat Sunhem. A comparison between shallow and deep architecture classifiers on small dataset. In 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), pages 1–6. IEEE, 2016.

[80] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. Journal of Big data, 3(1):1–40, 2016.

[81] Sadiq Alinsaif and Jochen Lang. Histological image classification using deep features and transfer learning. In 2020 17th Conference on Computer and Robot Vision (CRV), pages 101–108. IEEE, 2020.

[82] Dong Liu, Yaohui Liu, Shanglin Li, Weiqing Li, and Luda Wang. Fusion of handcrafted and deep features for medical image classification. In Journal of Physics: Conference Series, volume 1345, page 022052. IOP Publishing, 2019.

[83] Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. Machine Learning, 53(1):23–69, 2003.

[84] Gunjan Chugh, Shailender Kumar, and Nanhay Singh. Survey on machine learning and deep learning applications in breast cancer diagnosis. Cognitive Computation, 13(6):1451–1470, 2021.

[85] Frank E Grubbs. Procedures for detecting outlying observations in samples. Technometrics, 11(1):1–21, 1969.

[86] Rui Zhang, Shaoyan Zhang, Sethuraman Muthuraman, and Jianmin Jiang. One class support vector machine for anomaly detection in the communication network performance data. In Proceedings of the 5th Conference on Applied Electromagnetics, Wireless and Optical Communications, pages 31–37. Citeseer, 2007.

[87] Amit Banerjee, Philippe Burlina, and Chris Diehl. A support vector method for anomaly detection in hyperspectral imagery. IEEE Transactions on Geoscience and Remote Sensing, 44(8):2282–2291, 2006.

[88] Yungang Zhang, Bailing Zhang, Frans Coenen, Jimin Xiao, and Wenjin Lu. One-class kernel subspace ensemble for medical image classification. EURASIP Journal on Advances in Signal Processing, 2014(1):1–13, 2014.

[89] Yanan Wang, Litao Yang, Geoffrey I Webb, Zongyuan Ge, and Jiangning Song. Octid: a one-class learning-based python package for tumor image detection. Bioinformatics, 37(21):3986–3988, 2021.

[90] Long Gao, Lu Yang, Dooman Arefan, and Shandong Wu. One-class classification for highly imbalanced medical image data. In Medical Imaging 2020: Imaging

Informatics for Healthcare, Research, and Applications, volume 11318, pages 342–347. SPIE, 2020.

[91] Philipp Seeböck, Sebastian Waldstein, Sophie Klimscha, Bianca S Gerendas, René Donner, Thomas Schlegl, Ursula Schmidt-Erfurth, and Georg Langs. Identifying and categorizing anomalies in retinal imaging data. arXiv preprint arXiv:1612.00686, 2016.

[92] Tal Tlusty, Guy Amit, and Rami Ben-Ari. Unsupervised clustering of mammograms for outlier detection and breast density estimation. In 2018 24th International Conference on Pattern Recognition (ICPR), pages 3808–3813. IEEE, 2018.

[93] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11(12), 2010.

[94] Yu-Xing Tang, You-Bao Tang, Mei Han, Jing Xiao, and Ronald M Summers. Deep adversarial one-class learning for normal and abnormal chest radiograph classification. In Medical Imaging 2019: Computer-Aided Diagnosis, volume 10950, pages 305–311. SPIE, 2019.

[95] Haruna Watanabe, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Bone metastatic tumor detection based on anogan using ct images. In 2019 IEEE 1st Global Conference on Life Sciences and Technologies (LifeTech), pages 235–236. IEEE, 2019.

[96] Long Gao, Lei Zhang, Chang Liu, and Shandong Wu. Handling imbalanced medical image data: A deep-learning-based one-class classification approach. Artificial Intelligence in Medicine, 108:101935, 2020.

[97] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. ACM Computing Surveys (CSUR), 54(2):1–38, 2021.

[98] 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, pages 4393–4402. PMLR, 2018.

[99] Choubo Ding, Guansong Pang, and Chunhua Shen. Catching both gray and black swans: Open-set supervised anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7388–7398, 2022.

[100] Brian Kulis et al. Metric learning: A survey. Foundations and Trends® in Machine Learning, 5(4):287–364, 2013.

[101] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of Machine Learning Research, 9(11), 2008.

[102] Leif E Peterson. K-nearest neighbor. Scholarpedia, 4(2):1883, 2009.

[103] Liu Yang and Rong Jin. Distance metric learning: A comprehensive survey. Michigan State Universiy, 2(2):4, 2006.

[104] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 1, pages 539–546. IEEE, 2005.

[105] Yan Han, Chongyan Chen, Ahmed Tewfik, Ying Ding, and Yifan Peng. Pneumonia detection on chest x-ray using radiomic features and contrastive learning. In 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), pages 247–251. IEEE, 2021.

[106] Xiaocong Chen, Lina Yao, Tao Zhou, Jinming Dong, and Yu Zhang. Momentum contrastive learning for few-shot covid-19 diagnosis from chest ct images. Pattern Recognition, 113:107826, 2021.

[107] Saeed Shurrab and Rehab Duwairi. Self-supervised learning methods and applications in medical imaging analysis: A survey. arXiv preprint arXiv:2109.08685, 2021.

[108] Hari Sowrirajan, Jingbo Yang, Andrew Y Ng, and Pranav Rajpurkar. Moco pretraining improves representation and transferability of chest x-ray models. In Medical Imaging with Deep Learning, pages 728–744. PMLR, 2021.

[109] Anuroop Sriram, Matthew Muckley, Koustuv Sinha, Farah Shamout, Joelle Pineau, Krzysztof J Geras, Lea Azour, Yindalon Aphinyanaphongs, Nafissa Yakubova, and William Moore. Covid-19 prognosis via self-supervised representation learning and multi-image prediction. arXiv preprint arXiv:2101.04909, 2021.

[110] Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, et al. Big self-supervised models advance medical image classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3478–3488, 2021.

[111] Yichao Wu and Yufeng Liu. Robust truncated hinge loss support vector machines. Journal of the American Statistical Association, 102(479):974–983, 2007.

[112] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5659–5667, 2017.

[113] Shuyin Xia, Zhongyang Xiong, Yueguo Luo, Guanghua Zhang, et al. Effectiveness of the euclidean distance in high dimensional spaces. Optik, 126(24):5614–5619, 2015.

[114] Anna Jansen and E Andermann. Genetics of the polymicrogyria syndromes. Journal of Medical Genetics, 42(5):369–378, 2005.

[115] Nicolin Hainc, Mary Pat McAndrews, Taufik Valiante, Danielle M Andrade, Richard Wennberg, and Timo Krings. Imaging in medically refractory epilepsy at 3 tesla: a 13-year tertiary adult epilepsy center experience. Insights into Imaging, 13(1):1–9, 2022.

[116] Ekin Yagis, Alba G Seco De Herrera, and Luca Citi. Generalization performance of deep learning models in neurodegenerative disease classification. In 2019 IEEE

International Conference on Bioinformatics and Biomedicine (BIBM), pages 1692–1698. IEEE, 2019.

[117] Sadiq Alinsaif, Jochen Lang, Alzheimer's Disease Neuroimaging Initiative, et al. 3d shearlet-based descriptors combined with deep features for the classification of alzheimer's disease based on mri data. Computers in Biology and Medicine, 138:104879, 2021.

[118] Steven A Hicks, Inga Strümke, Vajira Thambawita, Malek Hammou, Michael A Riegler, Pål Halvorsen, and Sravanthi Parasa. On evaluation metrics for medical applications of artificial intelligence. Scientific Reports, 12(1):1–9, 2022.

[119] Jayawant N Mandrekar. Receiver operating characteristic curve in diagnostic test assessment. Journal of Thoracic Oncology, 5(9):1315–1316, 2010.

[120] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4):541–551, 1989.

[121] NV Shree and TNR Kumar. Identification and classification of brain tumor mri images with feature extraction using dwt and probabilistic neural network. brian informatics, 5, 23-30, 2018.

[122] Khaleda Akhter Sathi and Md Saiful Islam. Hybrid feature extraction based brain tumor classification using an artificial neural network. In 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), pages 155–160. IEEE, 2020.

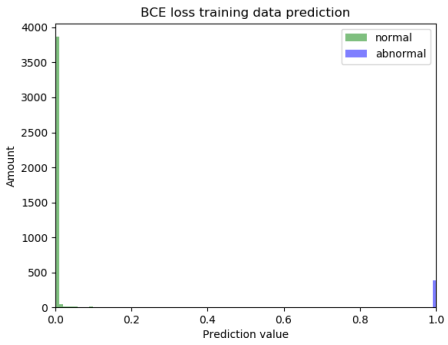
[123] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 2980–2988, 2017.

[124] Yu Zhou, Xiaomin Liang, Wei Zhang, Linrang Zhang, and Xing Song. Vae-based deep svdd for anomaly detection. Neurocomputing, 453:131–140, 2021.

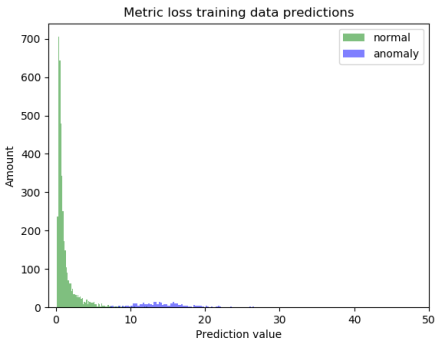
[125] Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics, 11(1):86–92, 1940.

[126] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In European Conference on Computer Vision, pages 630–645. Springer, 2016.

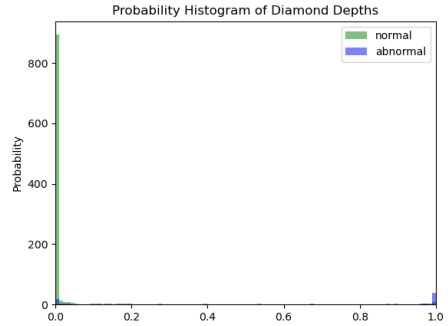
Appendix



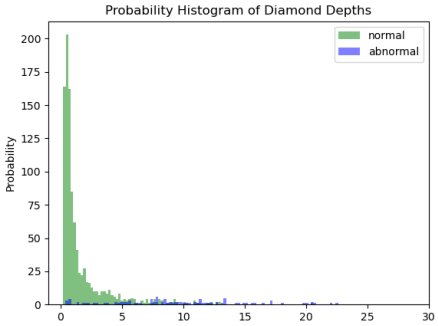
(a) BCE loss training



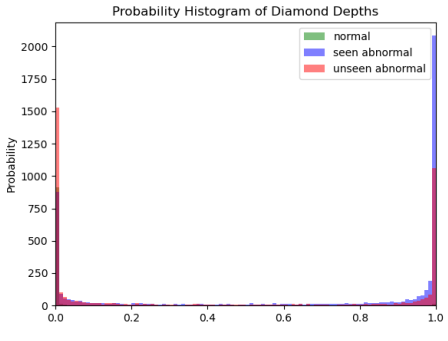
(d) our novel cDCM loss training



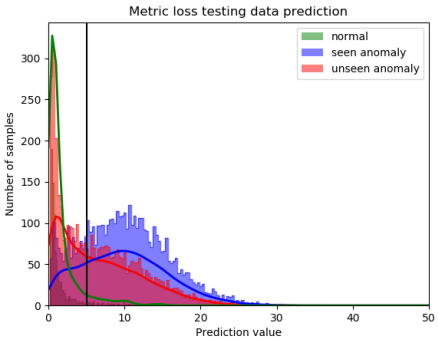
(b) BCE loss validation



(e) our novel cDCM loss validation

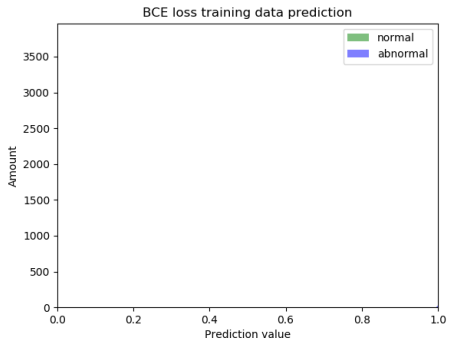


(c) BCE loss testing

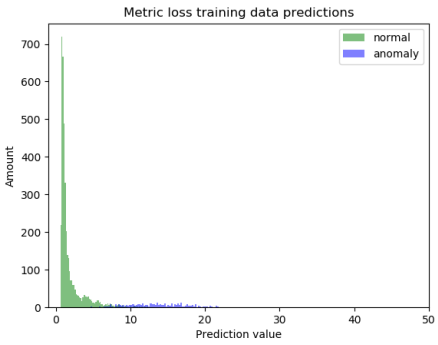


(f) our novel cDCM loss testing

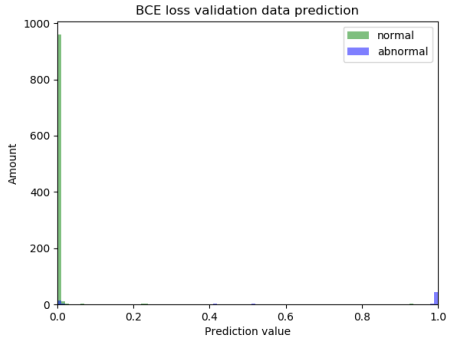
Figure 7.1: Modified CIFAR-10 data samples prediction distribution on BCE loss and our novel cDCM loss. Normal class is 0: airplane



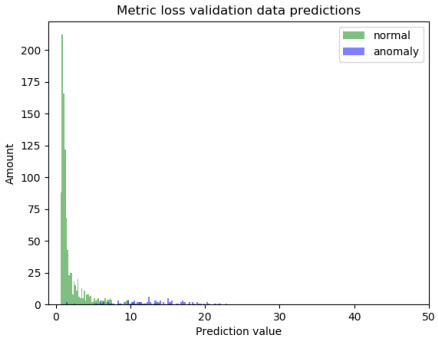
(a) BCE loss training



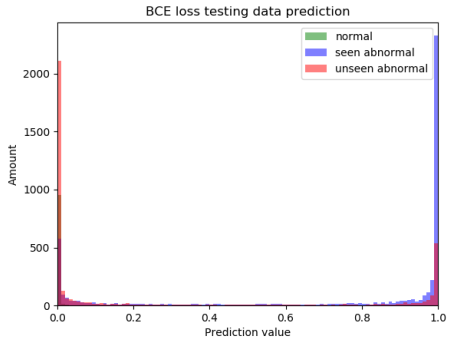
(d) our novel cDCM loss training



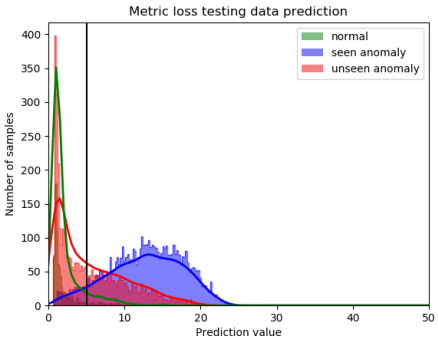
(b) BCE loss validation



(e) our novel cDCM loss validation

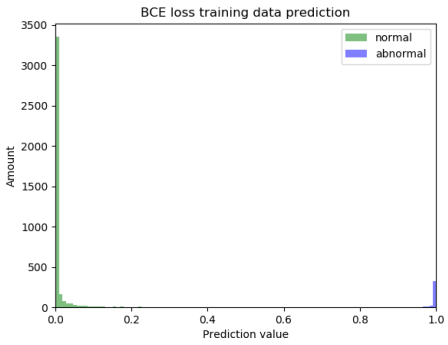


(c) BCE loss testing

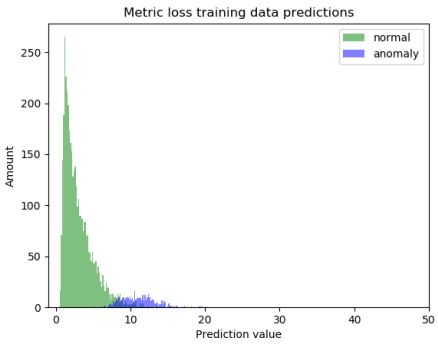


(f) our novel cDCM loss testing

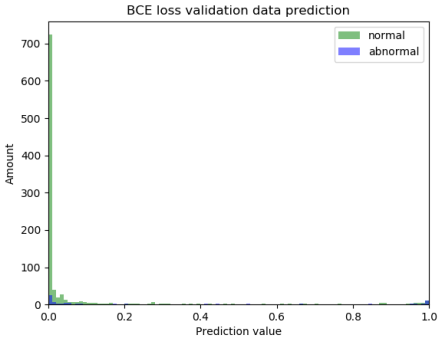
Figure 7.2: Modified CIFAR-10 data samples prediction distribution on BCE loss and our novel cDCM loss. Normal class is 1: automobile



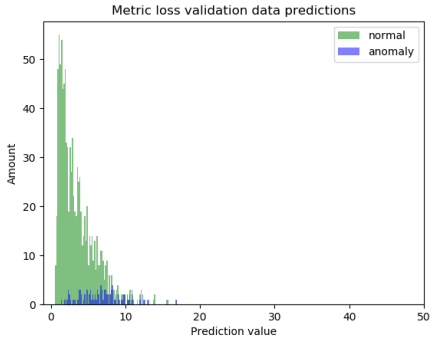
(a) BCE loss training



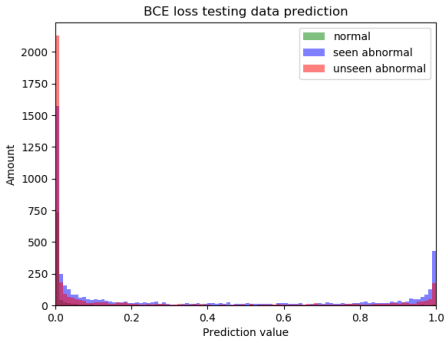
(d) our novel cDCM loss training



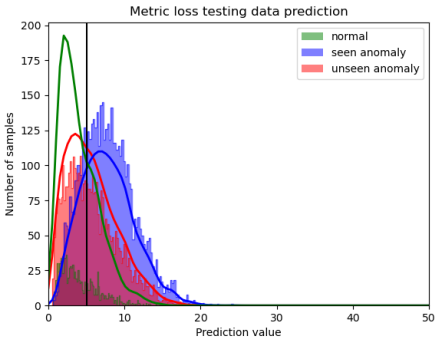
(b) BCE loss validation



(e) our novel cDCM loss validation

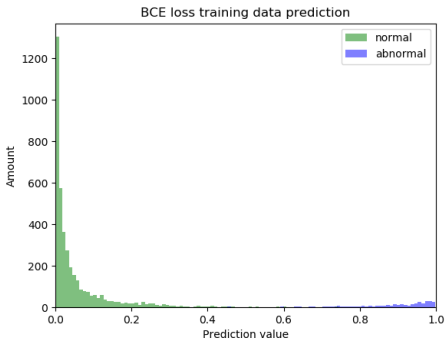


(c) BCE loss testing

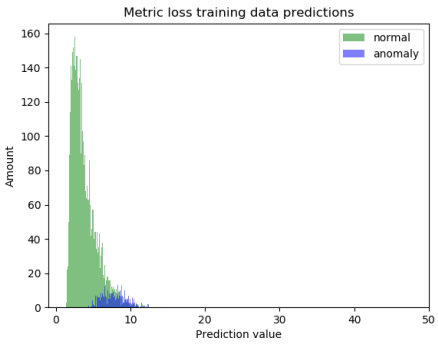


(f) our novel cDCM loss testing

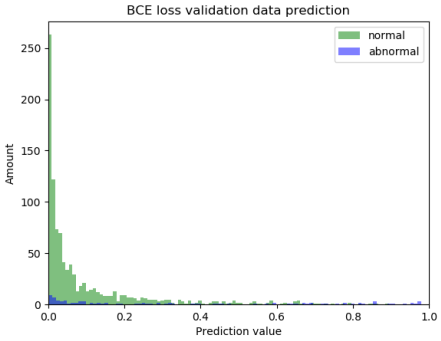
Figure 7.3: Modified CIFAR-10 data samples prediction distribution on BCE loss and our novel cDCM loss. Normal class is 2: bird



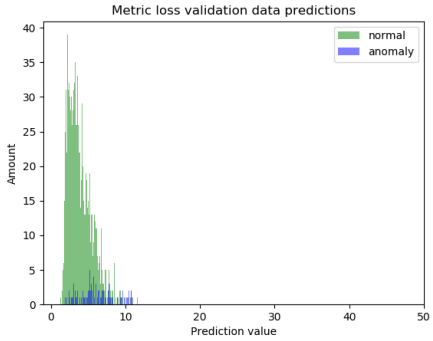
(a) BCE loss training



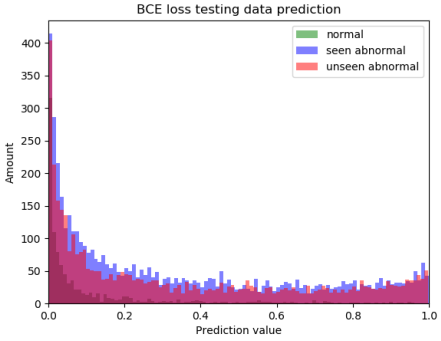
(d) our novel cDCM loss training



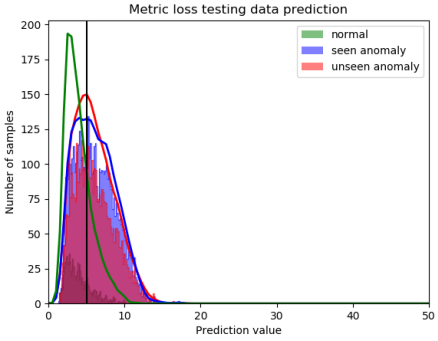
(b) BCE loss validation



(e) our novel cDCM loss validation

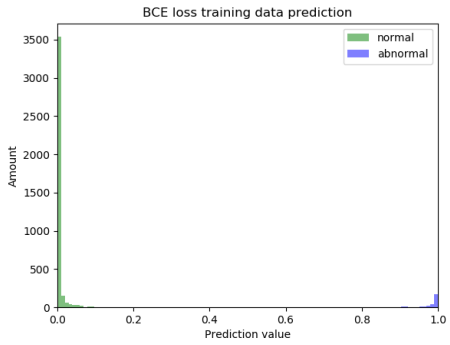


(c) BCE loss testing

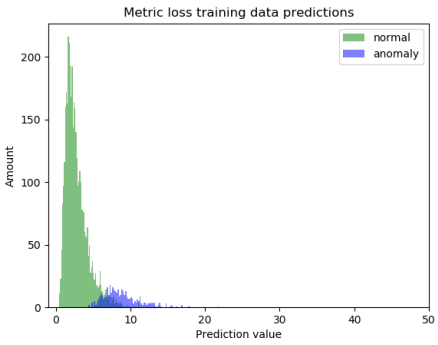


(f) our novel cDCM loss testing

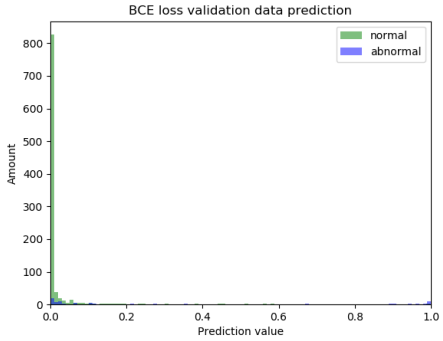
Figure 7.4: Modified CIFAR-10 data samples prediction distribution on BCE loss and our novel cDCM loss. Normal class is 3: cat



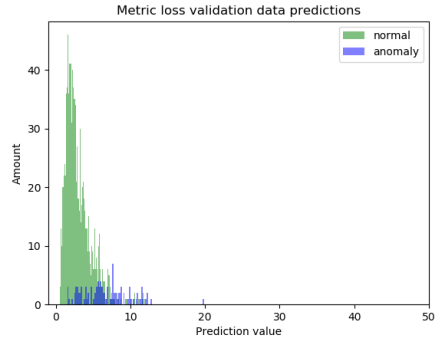
(a) BCE loss training



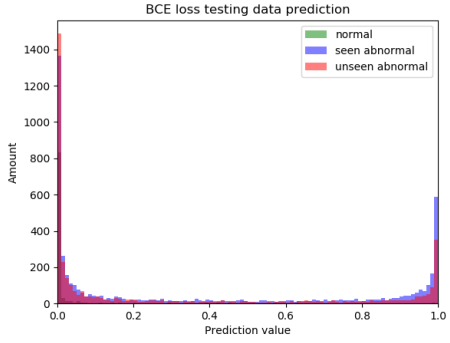
(d) our novel cDCM loss training



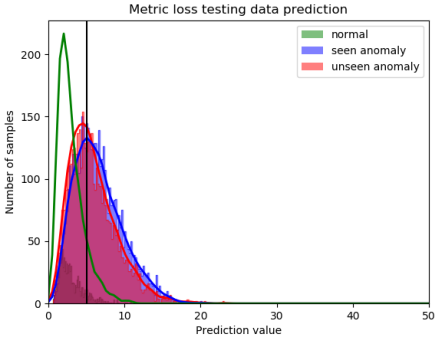
(b) BCE loss validation



(e) our novel cDCM loss validation

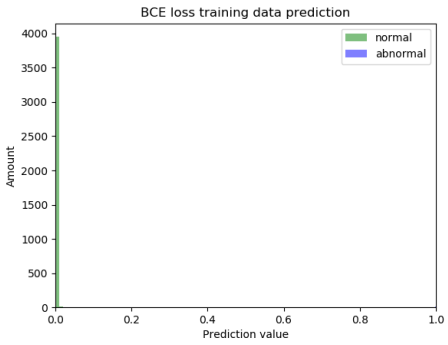


(c) BCE loss testing

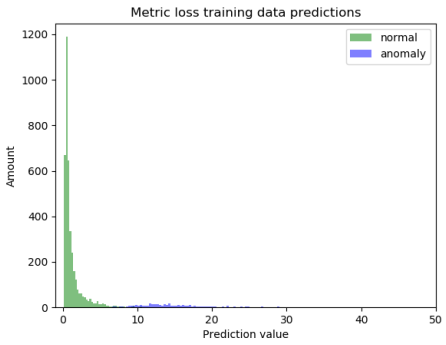


(f) our novel cDCM loss testing

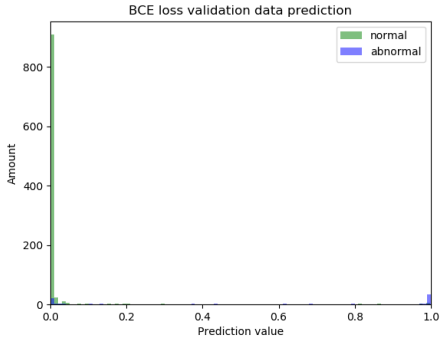
Figure 7.5: Modified CIFAR-10 data samples prediction distribution on BCE loss and our novel cDCM loss. Normal class is 4: deer



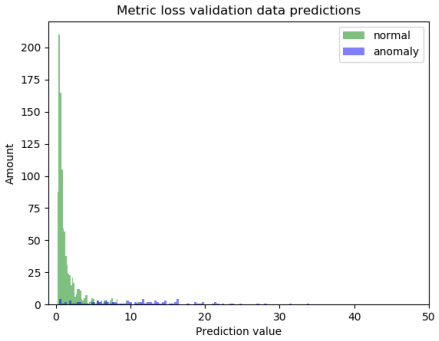
(a) BCE loss training



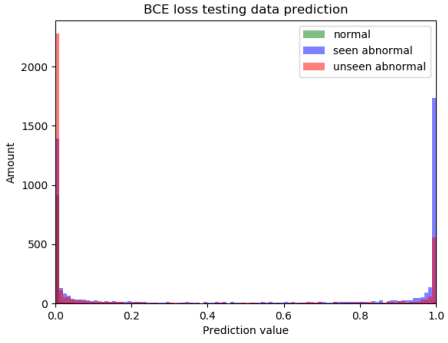
(d) our novel cDCM loss training



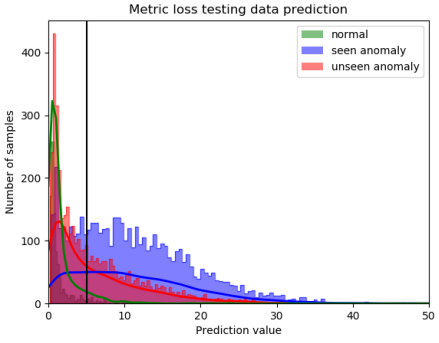
(b) BCE loss validation



(e) our novel cDCM loss validation

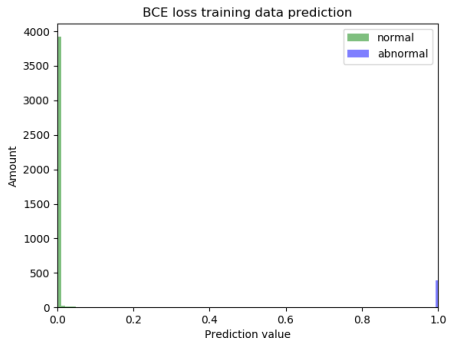


(c) BCE loss testing

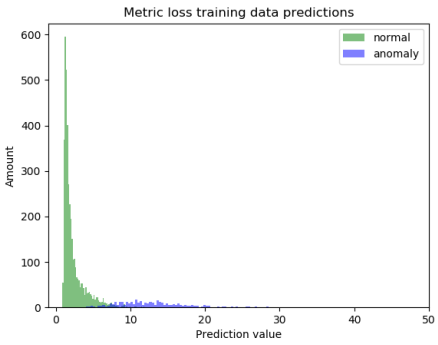


(f) our novel cDCM loss testing

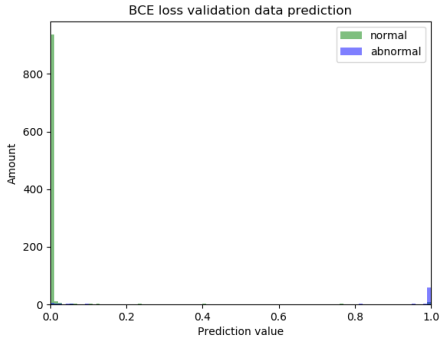
Figure 7.6: Modified CIFAR-10 data samples prediction distribution on BCE loss and our novel cDCM loss. Normal class is 5: dog



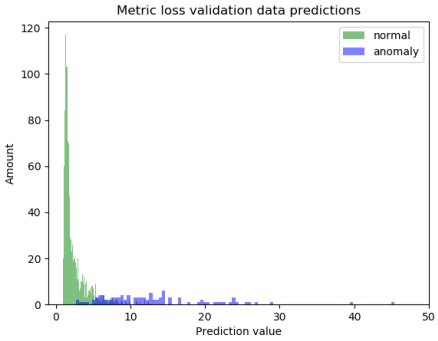
(a) BCE loss training



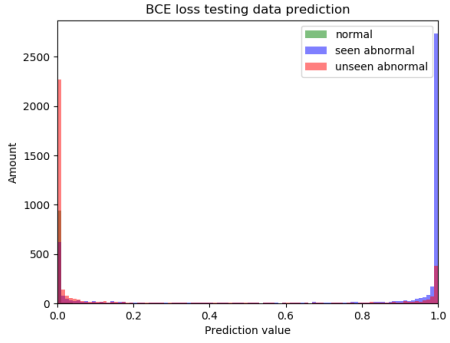
(d) our novel cDCM loss training



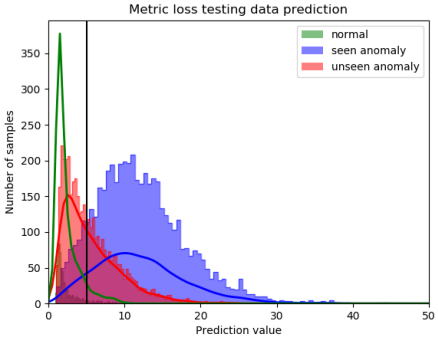
(b) BCE loss validation



(e) our novel cDCM loss validation

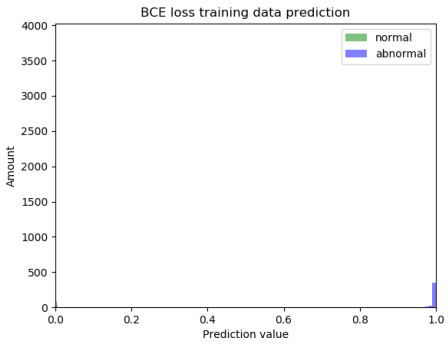


(c) BCE loss testing

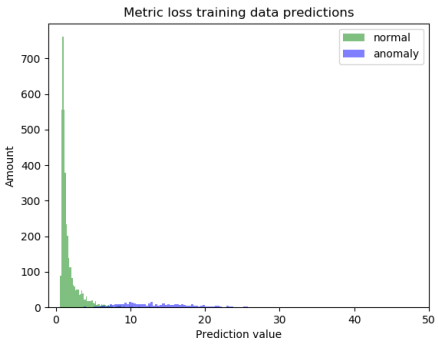


(f) our novel cDCM loss testing

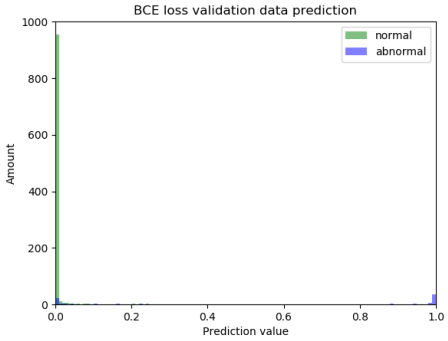
Figure 7.7: Modified CIFAR-10 data samples prediction distribution on BCE loss and our novel cDCM loss. Normal class is 6: frog



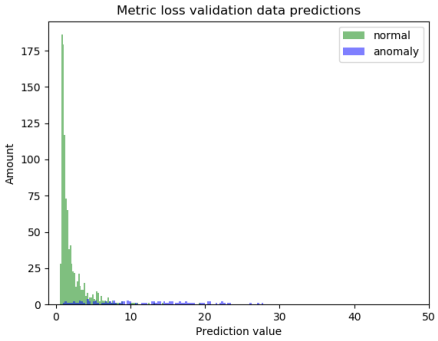
(a) BCE loss training



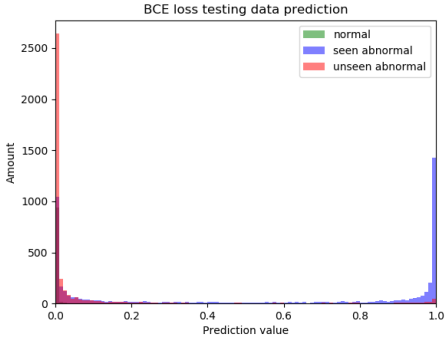
(d) our novel cDCM loss training



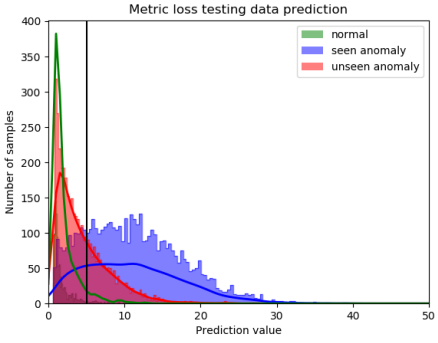
(b) BCE loss validation



(e) our novel cDCM loss validation

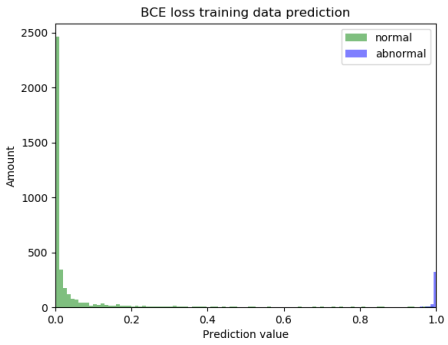


(c) BCE loss testing

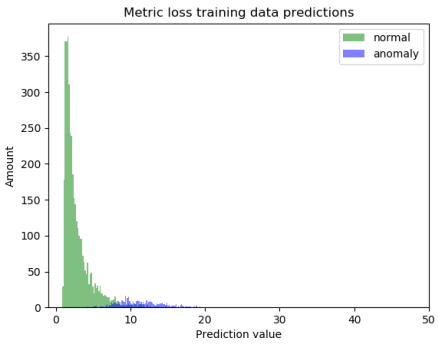


(f) our novel cDCM loss testing

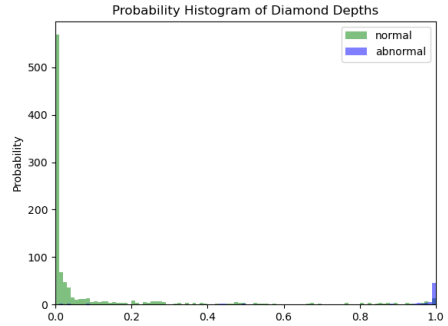
Figure 7.8: Modified CIFAR-10 data samples prediction distribution on BCE loss and our novel cDCM loss. Normal class is 7: horse



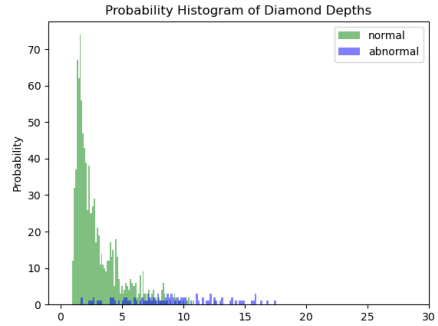
(a) BCE loss training



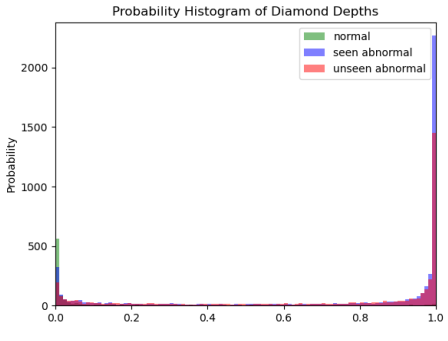
(d) our novel cDCM loss training



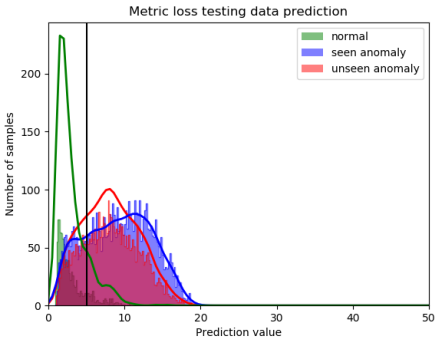
(b) BCE loss validation



(e) our novel cDCM loss validation



(c) BCE loss testing



(f) our novel cDCM loss testing

Figure 7.9: Modified CIFAR-10 data samples prediction distribution on BCE loss and our novel cDCM loss. Normal class is 9: truck