AN EMPIRICAL ANALYSIS OF CROSS-ENTROPY BASED AND METRIC-BASED METHODS ON NORTH ATLANTIC RIGHT WHALE ACOUSTIC DATA

by

Xuhui Liu

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
May 2020

Contents

List of Tables

List of Figures

Abstract

With increasing concern for marine species extinction, a massive effort has been made to conserve, prevent, and search for a sustainable solution. However, data labeling is a labor-heavy and time-consuming work, resulting in limited annotated acoustic data. What's more, a majority of labeled acoustic data are background noise. Both issues together raise interests in searching for solutions on how to effectively train a reliable classification model. We simulate different degrees of data compositions to study the impact of data scarcity and class imbalance on North Atlantic Right Whale (NARW) acoustic data. In the meantime, we explore two types of supervised deep learning approaches: metric-based classifiers and cross-entropy based classifiers. The empirical results show that our classifiers trained with fewer NARW acoustic data have comparable performance to the-state-of-art classifiers trained with a larger amount of acoustic data [1].

List of Abbreviations Used

PAM	Passive Acoustic Monitoring
ML	Machine Learning
DL	Deep Learning
DNN	Deep Neural Network
NARW	North Atlantic Right Whale
ONC	Ocean Network Canada
CV	Computer Vision
SNR	Signal to Noise Ratio
DCLDE	Detection, Classification, Localization, and Density Estimation of Marine Mammals
MARS	Marine Autonomous Recording Units
TLU	Threshold Logic Unit
CNN	Convolution Neural Network
LSVRC	Large Scale Visual Recognition Challenge
RESNET	Residual Neural Network
MSCOCO	Microsoft Common Object in Context
AP	Average Precision
PAC	Probably Approximately Correct
ROC	Receiver Operating Characteristic
PA	Positive Anchor

Acknowledgements

First, I would like to express my greatest gratitude to my supervisor, Dr. Stan Matwin, who gives me support, guidance, encouragement, and advice throughout my time here as a Masters student. I am incredibly grateful to you for supervising me and offering valuable opportunities to work on these exciting projects. Your understanding, consistent guidance and timely support contribute significantly to this thesis. I am so honored to have you as my supervisor.

Second, my sincere gratitude should go to my co-supervisor, Dr. Oliver Sølund Kirsebom, for your patient guidance, constructive advices, and encouragement. Thank you for spending your precious time in co-supervising me and guiding me throughout the development of this thesis. He has also spent a lot of time reviewing my paper.

Third, I would also like to thank Xiang Jiang, for his tremendous support and encouragement. He has also put a lot of effort into reviewing my thesis, and his feedback is crucial to me.

Next, I would extend my gratitude to my Meridian colleagues, Dr.Ines Hessler, Fabio Frazao, Bruno Padovese, and Matthew Smith, for their support and encouragement.

Finally, this research was enabled in part by support provided by Meridian (meridian.cs.dal.ca) and DeepSense (www.deepsense.ca).

I am thankful to all my friends and lab mates in the Big Data Institute and thank Dalhousie University for providing me such a friendly and quiet environment for my research work.

Thanks to my family and my fiancee for their love and support. Thank you all for the love and companionship.

Chapter 1

Introduction

In this chapter, we briefly present the current research problem, our motivations, approaches, contributions and the whole thesis outline. In Section 1.1, we introduce the motivation of conducting this research. We briefly describe the research objectives in Section 1.2 . Section 1.3 outlines the structure of this thesis.

1.1 Problem Setup

Increasing human activities, together with rising global warming issues, negatively affect the habitats of many species, reducing their quantity and diversity. Motivated by concerns of species extinction, considerable efforts are being devoted to conservation. For example, an extensive amount of labor is being put to monitor and conserve habitats [2], [3]. Marine animals are a subset of these endangered species; however, their different ecologies increase the difficulties of monitoring and surveying. More than 80% of marine species undertake a long migratory journey every year [4], for example, in search of food or to reach safe breeding grounds. Our focus, the North Atlantic Right Whale (Eubalaena glacialis), a cetacean type creature, is one of them. Because of their changeable migration pattern, low-cost monitoring and automated systems are preferred.

With the technology advances in recent decades, the Passive Acoustic Monitoring (PAM) system has been developed and applied in marine ecosystems [5], [6]. It has become a primary method for detecting and localizing marine animals owing to its cost-effective nature. A collection of hydrophone units constitutes the PAM system. It is deployed (in various ways) in the ocean to record sounds for days, weeks, even months. Such a process combines an automated or semi-automated computer program with human efforts in verification to detect the vocalization of marine animals in real-time or in the archived dataset. With expertise from experienced analysts, PAM can identify creature presence, vocal activity, and so on. [7]. Owing to the rapid development of the software and hardware, collecting and storing acoustic data has become more feasible than a few decades before. Sometimes, researchers can obtain more than a terabyte of data in a single project [8], [9]. With the growing volume of acoustic data, identifying and extracting vital ecological information becomes a bottleneck for human experts. In a conventional analytical process, human experts have to validate data visually and corroborate them aurally. Yet, this process is time-consuming and labor-heavy; but labelling the data is paramount for supervised machine learning (ML) tasks as the quality of labelled data strongly influences the performance of classification models.

Fortunately, these types of tasks can be handled effectively and consistently with the use of machine learning techniques [10]. With a fully trained ML model, human experts can execute the validation procedure automatically and in parallel, which substantially reduces the human effort. Despite some information is lost when transforming the audio signal to spectrogram, it is commonly used as the visual representation of the acoustic data and used by ML models in the automation of human expertise [5], [11], [12]. Popular ML methods that are applied in the acoustic domain include support vector machine [13], classifications and regressions trees [14], and recently deep learning [12], [15]–[17].

Deep learning (DL) is a subclass of machine learning algorithms and receives wide acclaim over classical machine learning approaches because of its superior performance in many complex tasks, for example, computer vision [18]. The convolutional neural network, which is an essential building block of most DL architectures, is evolved from traditional neural networks. These neural networks are partially inspired by the human brain and primarily consist of interconnected “neurons”. Each of these neurons corresponds to unique input data and is conditioned by specific weight. A simple linear function summarizes these inputs and bias as the new input for the non-linear activation function. This complete process happens in each layer of the neural network. With the use of an optimization algorithm in tuning weights and bias, it encourages the model to generate results closer to desired outputs. Shallow neural networks consist of a small number of layers, targeted at simple problems.

However, recent neural networks are based on deep architectures [19]. Such design enables the model to leverage stacked complex non-linear functions to search for the substantial information hidden in the dataset, hence unleashing the model's potential to better adapt to the given task. Thereby, models with deep architecture are commonly referred to as the Deep Neural Networks (DNN).

The success of DNN is attributed to the complex non-linear functions, carefully modeling the tasks, and the availability of a more substantial volume of datasets (ImageNet [20]). While lots of research publications focus on model performance on datasets with adequate examples, many recent research works on DL focus on how to train DNNs effectively on small datasets.

Though massive volumes of acoustic data have been collected systematically, the costly annotation process leads to minimal data being labelled. The strategy adopted by experts in this field is Transfer Learning [21], in which we first train our model in datasets from the same or different domains with abundant examples (millions of examples). After initial supervised training, models are capable of extracting key features from images and can adapt quickly to the task. Even though the model is trained on different datasets, enough training has enabled the model to achieve a compelling outcome.

1.2 Research Objectives

The research goal is to understand differences between cross-entropy based learning [22] and metric-based learning [23] in the context of the underwater acoustic classification task, and in the meantime, to provide insightful conclusions for effective training. To deal with this task, we follow the procedures below:

1. Survey existing methods that are applied to underwater acoustic classification task as well as relevant tasks.

2. Investigate differences between cross-entropy based learning and metric-based learning.

3. Evaluate the performance of cross-entropy based classifiers and metric-based classifiers trained with adjusted loss functions on NARW datasets.

4. List future improvements, and possible research directions of metric-based methods for the marine acoustic domain.

1.3 Thesis Outline

Chapter 2 explains the background of the acoustic data classification problem, the dataset used for this thesis, and the relationship between them. Additionally, the history of Deep Learning (DL) and different methods used in DL paradigms are briefly introduced. Chapter 3 introduces variants of cross-entropy and metric-based classifiers. Chapter 4 is the section for experiments, and we explain the model architecture and the alterations to the model. It also describes the experiment design, and the results and discussion section follows the experiment design. The last part of the thesis is Chapter 5, where the conclusions and future work are given.

Chapter 2

Background and Related Work

This chapter is organized as follows. Section 2.1 introduces the history of Passive Acoustic Monitoring (PAM) and the current research achievements using PAM. Section 2.2 describes the source of the dataset and how the data are processed. Section 2.3 introduces Deep Learning and compares two types of loss functions—i.e., cross entropy and metric-based losses—from the literature.

2.1 Passive Acoustic Monitoring

Passive Acoustic Monitoring (PAM) systems use underwater microphones (hydrophones) to detect, monitor, and, in certain cases, undertake the tasks of localizing, vocalizing marine mammals. Three types of passive acoustic equipment are used for capturing sounds: cabled hydrophones, autonomous recorders, and radio-linked hydrophones. The cabled hydrophone is typically deployed permanently or semi-permanently, and they are not in widespread use by academics, small organizations, and individuals due to expensive costs. There are a few organizations in Canada deploying these cabled hydrophone for data collection, for example, Ocean Networks Canada (ONC). ONC runs several world-leading observatories with the use of cabled hydrophones, and collects data on physical, chemical, biological and geological aspects of the ocean extensively [24]. On the other hand, autonomous recorders that consist of a hydrophone and battery-powered data-recording is comparatively affordable for research purpose. It is usually deployed in an array of three to ten instruments to offer regional coverage and sound source localization. The last type of recorder is the radio-linked hydrophone, which includes a hydrophone and radio link that connects with the ship or store station. Most recorders support internal data storage, which means they store collected data in the equipped disk [25]. A PAM system merely captures sounds incurred in the underwater environment and does not generate noise itself. Several characteristics [25] distinguish PAM from other monitoring methods, which are listed below,

1. long-term deployment,
2. immune to poor weather,
3. flexible deployment conditions (fixed or mobile).

Despite its applicability mentioned above, PAM still faces many challenges. The first issue is the level of ambient noise, which happens throughout data collection. Several factors contribute to this issue, including high-level shipping noise and fishing activities, which are part of anthropogenic sources, and also environmental factors like wind and precipitation events. This natural and anthropogenic noise complicates data analysis due to the low signal to noise ratio (SNR). The second problem results from variations in upcall (a stereotype contact call produced by NARW) patterns with respect to locations, seasons, time of the day, and genders of the species. The last issue is the similar sound produced by humpback whales (Megaptera novaeangliae), and this species has a much higher population than that of NARW. Moreover, these whales vocalize louder and more frequently, and also these two species are found co-occurred in the spring, sharing overlapping habitat and migratory routes.

2.1.1 Performance of Automated Detection or Classification Algorithms

An automatic detection or classification algorithm is necessary when analyzing a large volume of data because the benefits are fourfold [26]:

1. a computer never feels fatigue,
2. a computer is unbiased, or has a constant bias overall,
3. a computer algorithm can be executed distributively, ensuring the comparability of the results,
4. a computer works faster and can run in parallel, largely reducing processing time. For example, it runs the detection on right whale calls over five hydrophone-years of data within a week [27].

Thus, despite the fact that it has many challenges to be addressed, the detection and classification of NARW upcall using the PAM system has prevailed in the underwater bioacoustic domain. Spectrogram correlations with its parameters chosen manually and systematically are compared with the neural network, which is trained using backpropagation on 9/10 of the test dataset (consisting of 1857 upcalls or NARW and 6359 non-calls sounds), and the comparison result demonstrates the neural network is the best performing model, achieving less than 6% error rate [26]. Also, an automated detection system for right whale calls is developed with the synthetic kernel, targeted at exploiting the spectrogram cross-correlation. Though the detector has produced many false detections and missed individual calls, it indeed facilitates human analysts in their search of sections of data with a higher likelihood of having upcalls [28]. Another usage is found in Baumgartner et al. (2011), where they develop a detection and classification system to identify the low-frequency baleen whale calls [29]. Their system uses attribute extraction on pitch-tracking and combines a quadratic discriminant function analysis to detect and classify sei whale and NARW calls. Yu et al. compare traditional machine learning techniques with deep neural networks, and found deep neural networks on average achieve higher precision and recall. For instance, while the best detector of DCLDE 2013 got 65% retrieval rate of upcalls, deep neural network achieves 85% to over 90% retrieval rate on the DCLDE 2013 NARW validation datesets [1]. Also in the same paper, the best results that have been obtained on the DCLDE 2013 dataset using large training datasets are achieved by LeNet [30] and BirdNet [31] with average precisions of 0.903 and 0.891, respectively. Other usages [32], [33] can be found in the literature. We use the LeNet and BirdNet as described above as baselines in our work.

2.2 North Atlantic Right Whale

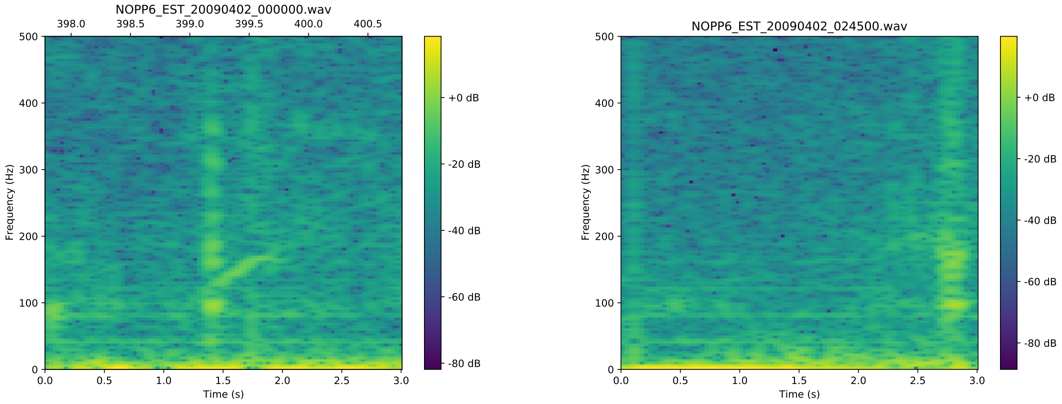
North Atlantic right whales (Eubalaena glacialis) are cross-listed as an endangered animal in Canada and the U.S. [34]. They are a species of cetacean and have up to 400 individuals estimated alive. They used to occupy an area along the U.S. continental shelf, reaching the Bay of Fundy and the Western Scotian shelf in Canada. However, they have changed their habitat in the last decade and have recently been observed active in the western Atlantic Ocean from southern Greenland and the Gulf of St.

Lawrence south to Florida. Although most trajectories are identified, the occurrence and movement within some areas remain unknown to researchers; for example, the waters off the U.S. coast between Georgia and Cape Cod. So far, with regulations and policies from the management authorities, their population has slightly recovered but has suffered a decline in recent years. In 2017, an unusual mortality event caused the loss of 12 individuals, again depressing the population of this species.

The North Atlantic Right whale can produce various vocalizations, but the upcalls are a typical proxy measure of the species' presence, which is characterized by an upsweep frequency from 50 to 350 Hz [35]. This stereotyped contact call, about a second in duration, is frequently used as a target for detection and classification systems.

2.2.1 Train and Test Datasets

In this thesis, we use the dataset provided in the workshop on Detection, Classification, Localization, and Density Estimation of Marine Mammals (DCLDE 2013). The PAM data are collected in 2000, 2008, and 2009 off the coast of Massachusetts with Cornell Marine Autonomous Recording Units (MARS). These collected data have been manually analyzed by marine experts, who have identified and labelled all occurring upcalls in the dataset. The raw recordings from these units are processed and cleaned for the detection and classification tasks. This published dataset (see Table 2.1) contains upsweeps calls from the right whale over seven days. As many publications prefer spectrograms [1], [32], [33] due to direct representational correlation between time and frequency, we thus convert a subset of DCLDE data to spectrograms for our experiments. We use the recommended parameter settings [5] to generate a spectrogram with a resolution of 94×129, representing time in the horizontal axis and frequency in the vertical axis, respectively. Each spectrogram representation is of a 3-second segment with frequency ranging from 0 to 500 Hz computed using a window size of 0.256s, a step size of 0.032s, and a Hamming window. We process these data using the Ketos package [36]. The examples of resulting spectrograms (in 94×129 resolution) are visualized in Figure 2.1, where Figure 2.1.(a) has an upcall displayed between 40 and 60, while Figure 2.1.(b) does not.



(a) *Positive example*

(b) *Negative example*

Figure 2.1: *(a) Spectrogram containing upcall signal (a short curve starting from 1.0s to 2.0s). (b) Spectrogram containing background noise. (These two figures are generated by Ketos package [36].)*

2.3 Deep Learning for Supervised Image Classification

This section provides the background of deep learning in the context of supervised image classification, and three major components constitute typical machine learning problems. Section 2.3.1 introduces the basic concepts of deep learning, followed by cross-entropy and metric-based loss functions in Section 2.3.2.

2.3.1 Deep Learning

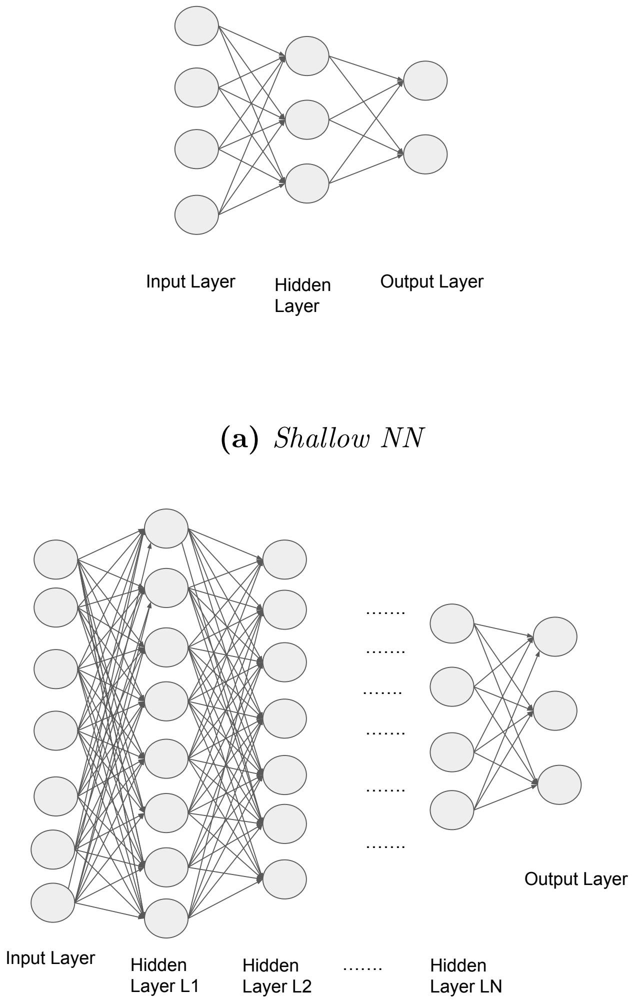
The idea of Neural Network is first introduced by Warren McCulloch and Walter Pitts in 1943 when they developed a technique known as the "threshold logic unit" to mimic the way the neuron was thought to work. This Threshold Logic Unit (TLU) was later given different terminologies, for example, Linear threshold unit, perceptron, and artificial neuron. Though perception (linear model) demonstrates reliable performance in tasks like linear regression and logistic regression, it has apparent limitations, for example, linear separability [22]. A neural network is called a network because it consists of many different functions to express complex functionalities. It is typically associated with a directed acyclic graph illustrating how functions are stacked and worked together. It has three fundamental components, an input layer,

	Recording date	Region	Total Recording Hours	Number of Upcalls
DCLDE 2013 workshop	28-Mar-09	Massachusetts	24	767
	29-Mar-09	Massachusetts	24	2,280
	30-Mar-09	Massachusetts	24	1,663
	31-Mar-09	Massachusetts	24	2,206
	1-Apr-09	Massachusetts	24	1,328
	2-Apr-09	Massachusetts	24	545
	3-Apr-09	Massachusetts	24	894
Train dataset	Sampled from 2-Apr-09 to 3-Apr-09	Massachusetts	-	1024
Evaluation dataset	Sampled from 28-Mar-09	Massachusetts	-	512

Table 2.1: Data Sources of original dataset, training dataset and evaluation dataset. Number of upcalls is the number of upcalls identified by the trained analyst.

an output layer, and some hidden layers residing between the input layer and the output layer (see Figure 2.2). These chained layers assemble the model from end to end. The overall length of this chain refers to the depth of the model. It is the reason why the model is named Deep Neural Network (DNN), and such a model learning process is known as "deep learning" (DL). The model accepts the input x, outputs a result y, and information flows forward in between. The information from x is propagated to each hidden layer and finally reaches the output layer to produces y. We note such a feed-forward process as forward-propagation, and it updates the results of each layer. The second milestone occurs when backpropagation [37] is proposed, which allows for the reversed propagation of weight adjustment information. Early success is observed in training a convolution neural network (CNN) to recognize handwritten digits. The model is known as LeNet, which was developed by Yan Lecun [30].

CNN is a specialized neural network designed for handing data with grid-like topology, especially time-series data and image data. Unlike the standard neural networks that employ the matrix multiplication in layers, CNN uses a "convolution" operation to downsample the data, resulting in a noticeable decrease of parameters. It further promotes computation efficiency and memory efficacy. Convolution operations



(a) *Shallow NN*

(b) *Deep NN*

Figure 2.2: *(a) Illustration of a shallow NN architecture. (b) Illustration of a Deep NN architecture.*

leverage three critical ideas that are important in improving model performance, including sparse connectivity, parameter sharing, and equivariant [22]. Traditional NN computes the result of a subsequent layer by performing a matrix multiplication of current input units, and preceding output units. It thus requires one-to-one interaction between units in these two consecutive layers. This operation is costly and sometimes not necessary. CNN, therefore, uses a small-size kernel to downsample the data. It is inspired by how the visual mechanism works in the brain, where each cell in the visual cortex is responsible for corresponding receptive fields. By doing downsampling, it means to extract small, meaningful features, such as edges, from thousands of pixels. This operation significantly reduces the number of parameters, leading to less memory cost and a noticeable improvement in statistical efficiency. Parameter sharing also plays a vital role in boosting training efficiency and reducing model size. It refers to using the same kernel parameters repeatedly in the same layer. With parameter sharing, each layer, therefore, has a property called equivariance to translation, meaning the output changes in the same way as the input changes. The pooling layer comes as the last component in CNN architecture and typically follows the convolution layer and activation function. Max pooling is the most popular pooling function employed in the pooling layer, which picks the maximum value within a specific range as the resulting output. The pooling function summarizes the output of the net with values of the highest statistical significance. Since then, the DNN has been capable of resolving complex tasks. A notable achievement was witnessed in 2012 when DNN surpassed traditional machine learning methods and won the Large Scale Visual Recognition Challenge (LSVRC) [38].

2.3.2 Representation, Optimization and Evaluation

A machine learning problem can be decomposed into three components: representation, optimization and evaluation [39]. In the context of deep learning, representation is defined by the compositions of various differentiable functions, and learning is achieved by gradient-based optimization of model parameters. We evaluate the model's performance through a held-out set from the same distribution of the training examples.

For supervised deep learning methods, there are at least two approaches to formulating the representation of a model: non-parametric metric-based and parametric classifiers. The non-parametric method is similar to K-nearest neighbor learning where class labels are assigned according to distance-based inference, without the need to optimize a parameterized classifier. The goal of metric-based learning is to learn a mapping function from the input space to a metric space where distances between examples are semantically meaningful (the same-class instances are clustered and different-class instances are separable). The learning procedure can be interpreted as an optimization step that promotes margin maximization and class compactness. A parametric classifier, on the contrary, typically learns a deterministic function from the input space to the label space through direct optimization of model parameters using the cross-entropy between the ground-truth labels and the model's probabilistic prediction for each class.

In terms of evaluation, there are a number of assessment metrics: precision, recall, f1-score and accuracy, to name a few. In binary classification task, there are four basic combinations of actual data category and assigned category: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Precision measures the classifier's ability in not labeling negative examples as positive examples (see Eq. 2.1) while the recall assesses the classifier's ability in retrieving all positive samples (see Eq. 2.2). The F1-score can be interpreted as the weighted average of precision and recall, therefore the relative contribution of precision and recall to F1-score are equivalent. Accuracy (see Eq.2.4), to be specific, the classification accuracy is the rate of correct classifications.

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.3}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.4}$$

In addition to the general concept of two approaches, a nontrivial factor in the implementation phase that controls the stability and speed of the model training process can not be ignored, which is batch size. Batch size defines the number of training examples sampled from the training dataset and used in the estimate of error gradient (in parametric classifiers) or distance (in non-parametric classifiers). The batch gradient descent method uses the entire set of training samples to compute the gradient each time, whereas the stochastic gradient descent approach exploits one example at a time [22]. However these two approaches have their limitations, for instance, batch gradient descent suffers from slower and harder optimization because of its deterministic nature, and stochastic gradient descent faces oscillation issues with noisy data. Thus, mini-batch gradient descent is proposed to mitigate the above mentioned issues. Its size is smaller than the entire dataset but larger than one [22].

2.4 Learning with Class imbalance

As described in Section 2.1, the class-imbalance problem is often present in the PAM dataset, and it is a critical issue that has been widely studied in the literature [40]–[50]. Imbalance tends to severely impair the performance of classifiers by ignoring the minority classes during the training phase [40], [49]. There are three main types of approach, including re-sampling strategies, changing samples' importance, and tree-based ensemble learning. Under-sampling (sometimes referred to as down-sampling) and over-sampling (up-sampling) [41] are two exemplary implementations of re-sampling algorithms [49], [50] (other examples include NearMiss [47], One-sided selection [45] , SMOTE[51] and etc.). They involve a bias to select more samples from one class than another to compensate for the imbalance presented in the data. For example, Oquab et al. [48] re-sample the foreground and background image patches in their work. However, the cost of misclassifying majority class samples is nontrivial, and it often results from an under-sampled majority class. When under-sampling the dataset, one intrinsically hypothesizes that the cost of misclassification of these classes is similar, but that might be wrong. Therefore, Elkan et al. [42] propose cost-sensitive learning to heavily penalize the wrong classification of a minority class. The final approach is through ensemble-learning based methods, which typically incorporate tree-based algorithms. For instance, Liu et al. [46] propose EasyEnsemble and BalanceCascade. The EasyEnsemble algorithm ensembles the Adaboost classifiers trained with data consisting of non-overlapping subsets of majority class instances and repeated minority data. BalanceCascade iteratively removes the correctly classified examples to reduce the redundant information in the majority class.

In this thesis, we use a type of over-sampling approach, which we call "class-balanced-sampling" that makes use of the stochastic nature of gradient-based optimization where the model is required to sample a large number of stochastic batches for each optimization step. With the class-balanced-sampling strategy, we draw uniform samples from different classes to construct each batch such that the model perceives a balanced empirical distribution. In essence, each batch uses under-sampling to simulate balanced examples while all batches together achieve over-sampling of the minority class.

2.5 Transfer Learning

The costly annotation work results in limited labeled data available for training models; however, the amount of labeled data largely influences the model's capability. One way to address the data scarcity issue is to use Transfer Learning. Transfer learning (TL) is a popular research problem that focuses on applying learned knowledge from the one domain to the same or different domain [21]. Some applications [52], [53] take advantage of learned knowledge and apply them to acoustic data.

Another interesting strategy is by using few-shot learning to address the data scarcity. Unlike some DNN models have access to abundant labeled data, the bottlenecks for many real-world applications are the shortage of annotated data. Thus, it raises interests in searching for solutions to generalize the model to classify unseen classes with limited samples per novel class. The few-shot learning [54]–[60] considers using fewer data samples per class along with gradient-based or metric-based fine-tuning to adapt classifiers to unseen classes. The typical few-shot learning problem has the form of C-way K-shot, where a fixed number C stands for unique classes used to train the classifier and K is the number of samples per class. While the mainstream focus of few-shot learning is on Computer Vision (CV), several few-shot learning based methods have been applied to acoustic data. Chou et al. introduced an attentional similarity module to several metric-based learning methods, and they demonstrated consistent improvement for all the tasks of few-shot sound recognition[55]. Wang et al. used the Prototypical Network, a metric-based few-shot learning method, to detect the similar-sounding events [60]. Shimada et al. trained metric-based few-shot learning methods to clearly separate the background noise from other event sounds and also to detect the rare sound events [57]. Xiang et al. extended the few-shot learning to few-to-medium-shot learning and adopted a new learning procedure [56], where all training data are used as the embedding to help with the classification in the evaluation phase. Our NARW dataset contains two classes ($C = 2$), which is different from the aforementioned few-shot learning scenarios that have sufficient unique classes to be sampled each time during the training. Therefore, the few-shot learning is not applicable in our case.

Chapter 3

Cross-entropy and Metric-based Image Classification

Section 3.1 introduces three variants of Cross-entropy methods. It is followed by an introduction of metric-based methods (Section 3.2), including variants of the Contrastive loss and variants of the Triplet loss. Section 3.3 compares the differences between these two types of methods.

3.1 Cross-Entropy Based Learning

Cross-entropy loss is widely used in machine learning classification and optimization tasks [38], [61]. Our task is a binary classification with class labels 0 and 1, denoting the negative and positive labels. The formula of a binary cross-entropy loss is defined below (see Eq. 3.1), where y_0 and y_1 are the ground-truth labels, and $\hat{y_0}$ and $\hat{y_1}$ are predicted labels. In the training stage, the Softmax function assigns training examples with the labels having the highest probability. Consequently, when the instance is assigned a wrong label, the cross-entropy loss function tries to minimize the probability of the wrong label but, in the meantime, increases the probability of the correct label. It iteratively ensures the predicted labels of training data match their ground-truth labels.

$$L_{cross-entropy} = y_0 \times \log(p(\hat{y_0})) + y_1 \times \log(p(\hat{y_1})) \tag{3.1}$$

Cross-entropy can be decomposed into the entropy of the ground-truth labels ($H(p)$) and the Kullback–Leibler divergence $D_{KL}(p \parallel q)$ of the predicted model distribution q from the ground-truth distribution p. It is worthwhile to mention that cross-entropy is calculated at the example-level where the predicted model distribution can be understood as probabilities for assigning an example x into different classes $y_1, \ldots, y_N$, rather than the empirical distribution of all training examples; therefore, instance-level sampling does not interfere with cross-entropy minimization (see Eq. 3.2).

$$J(w) = \frac{1}{N}\sum_{n=1}^{N} H(p,q) = -\frac{1}{N}\sum_{n=1}^{N}[y_n \times \log(y_n) + (1-y_n) \times \log(1-y_n)] \qquad (3.2)$$

3.1.1 Cross-Entropy with Random-Sampling

With random-sampling, the class percentage in each batch follows the empirical class distribution of the data. We visualize it in Figure 3.1(b)

3.1.2 Cross-Entropy with Over-Sampling

We introduce the over-sampling strategy into the cross-entropy loss function to compensate for the data scarcity of the minority class in imbalanced datasets. This over-sampling samples one negative instance when every N positive is presented in the batch (1:N ratio). In Figure 3.1(c), we use the 1:9 ratio to sample negative and positive samples.

3.1.3 Cross-Entropy with Class-Balanced-Sampling

The class-balanced-sampling is a variant of the over-sampling strategy with equal sample rates of both classes. In the class-balanced-sampling strategy, we sample the same number of examples from each class so as to have equal contributions from both the majority and the minority class while optimizing model parameters (see Figure 3.1(a)).

3.2 Metric-based Learning

Section 3.2.1 introduces the variants of standard "contrastive loss", which implements a different sampling strategy and the subsequent section 3.2.2 discusses variants of "Triplet loss".

In metric-based classifiers, the input x is typically fed into the model, and the output is mapped into a large manifold space, sometimes referred to as embedding space. The embedding space varies in dimensionalities, from lower dimension 2 to

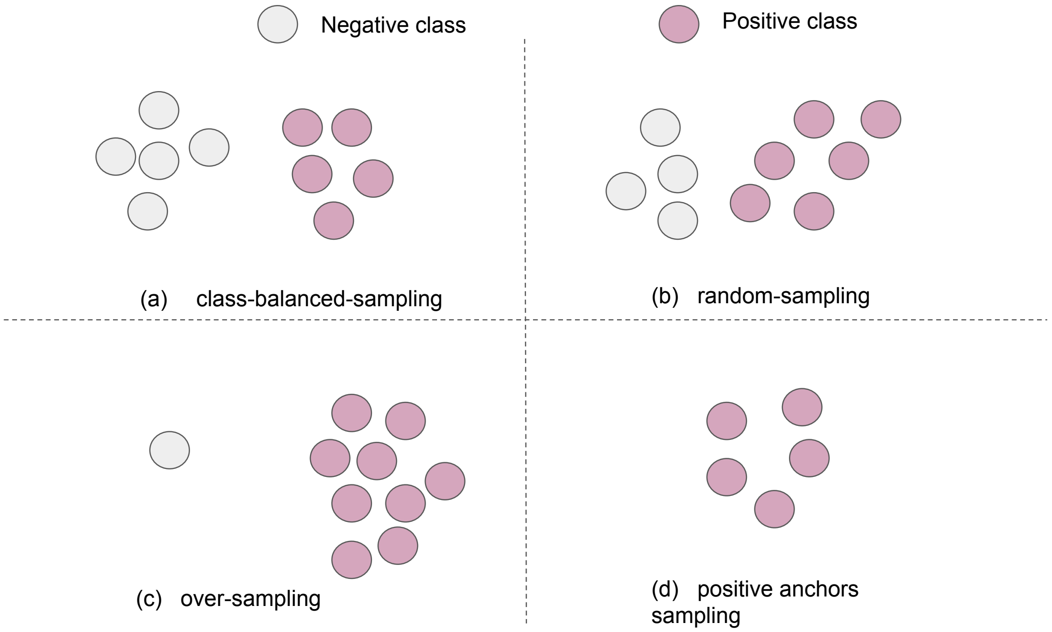


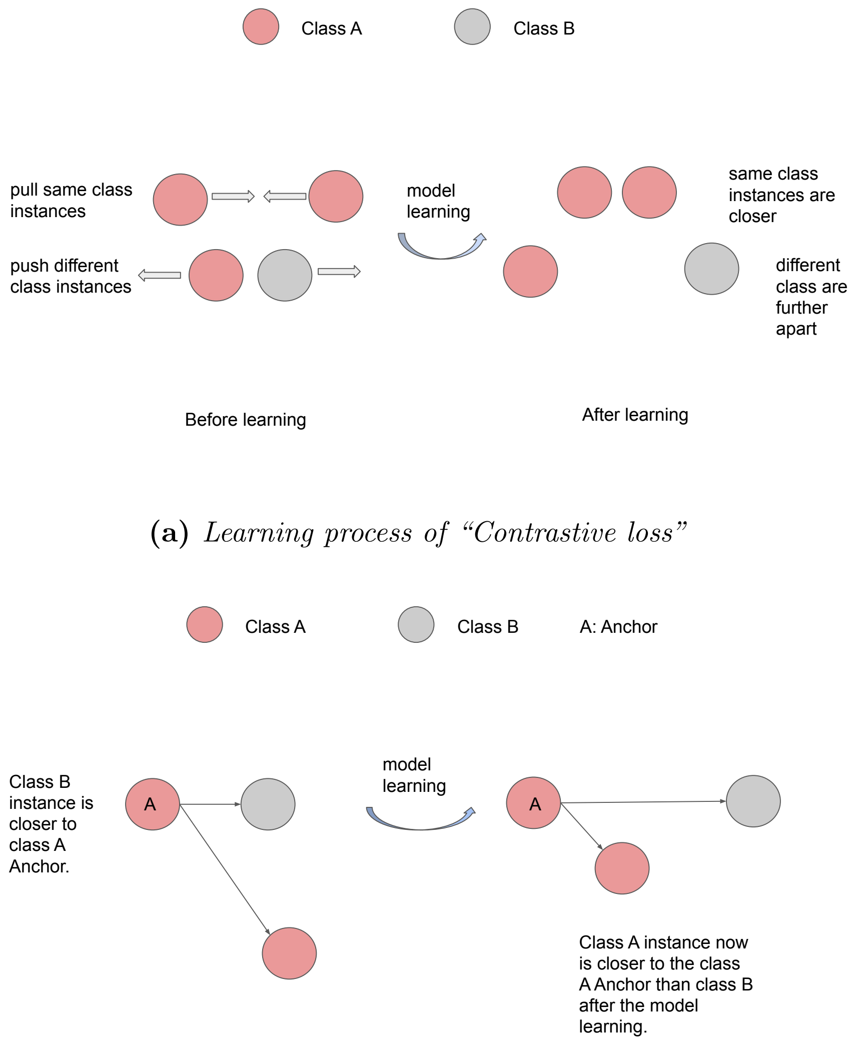
Figure 3.1: *Batch sampling strategies.*

higher dimension 512 (might be higher). Metric-based classifiers (we use K-Nearest Neighbors with $K = 5$) classify the test instance based on its similarity to other representative training data in the embedding space. The similarity is commonly measured via the distance between data points, and the Euclidean distance is the widely employed distance function (see Eq. 3.3), where p and q are two different points in the metric space. A semantically meaningful embedding space is the key to the success of metric-based classifiers. It is obtainable through metric-based learning to maximize the distance of the inter-class examples and, in parallel, to minimize the distance of the intra-class examples. Two classic metric-based loss functions in the literature are Contrastive loss [62] and Triplet loss [63].

$$d(p, q) = ||p - q|| \tag{3.3}$$

3.2.1 Contrastive Loss Function with Different Sampling Strategy

The design of "Contrastive loss" considers two conceptual classes, the same class, and a different class. It pulls the same-class instances into the same cluster and, in the meantime, pushes different-class examples apart. Its formula is shown below (see Eq. 3.4 [62]), where x_i and x_j are projected data instances in the metric space, y


Class A
Class B
pull same class instances
push different class instances
model learning
same class instances are closer
different class are further apart
Before learning
After learning
Class A
Class B
A: Anchor
Class B instance is closer to class A Anchor.
A
model learning
A
Class A instance now is closer to the class A Anchor than class B after the model learning.

(a) *Learning process of "Contrastive loss"*

(b) *Learning process of "Triplet loss"*

Figure 3.2: *(a) Illustration of "Contrastive loss" function learning. (b) Illustration of "Triplet loss" function learning.*

denotes whether x_i and x_j are from the same class, and $m > 0$ stands for a margin. m is used to maintain a certain distance between different classes. Figure 3.2.(a) illustrates the complete learning process.

$$L(y, x_i, x_j) = \frac{1}{2}(1 - y)d(x_i, x_j) + \frac{1}{2}y \max(0, m - d(x_i, x_j)) \tag{3.4}$$

We think that there is a drawback of such a design, especially when being used in the imbalanced dataset. For example, the pull loss remains the same; in contrast, the push loss treats those minority data as outliers (or noise) and ignores their contributions to the weight's tuning. Therefore, this process turns the model's prediction inclining to the classes with the majority data. Consequently, it leads to outstanding predictability for these classes and bad predictions for classes with fewer data. To mitigate these issues, and also motivated by the resampling [48], [64], we introduce class-balanced-sampling, over-sampling, and also positive-only sampling strategies into the standard "Contrastive loss", forming two groups of variants,

1. Contrastive loss based
2. Positive Pair based

The variants of "Contrastive loss" function still use the original formula while the "Positive Pair loss" function reformulates as Eq. 3.5, where x_i and x_j are two different projected data instances in the metric space.

$$L(y, x_i, x_j) = \frac{1}{2} y d(x_i, x_j) \tag{3.5}$$

Contrastive loss with random-sampling. The base case uses random-sampling, as illustrated in Figure 3.2b. The ratio of positive samples contained in the batch is random from batch to batch.

Contrastive loss with over-sampling. The acoustic dataset might contain sizeable negative examples (background noise) and less considerable positive examples. It requires a method either to balance examples from two classes or to over-sample the minority class. We thus put a high sampling ratio to the minority class, for instance, 90% (visualized in Figure 3.1(c)). This over-sampling approach is taken to compensate for the insufficient samples from the minority class.

Contrastive loss with class-balanced-sampling. We adopt the class-balanced-sampling strategy (see Figure 3.1(a)) in the "Contrastive loss" to balance the examples from each class when forming the batch. This process has the model to pay equal attention to both classes. Ideally, these classes contribute equally to the weights' update.

Positive Pair Loss. As aforementioned, "Contrastive loss" considers two losses, the pull loss, and the push loss. The pull loss enforces the class compactness in the embedding space while the push loss expands the class-to-class distance. We assume the push loss might harm the prediction of the minority class. Hence, we attempt to remove the contribution of the push loss during model training and take only the pull loss into account. It forms the "Positive Pair loss", which considers the interaction between instances from the same class. We sample the same amount of samples from both classes (Figure 3.1(a)) and apply the above loss function to compute the pull loss.

Positive Pair Positive Anchor Loss. In addition to the previous modification to the standard "Contrastive loss", we again attempt to adjust the "Positive Pair loss" to become "Positive Pair Positive Anchor loss." As the name suggests, this loss function completely ignores the effect of the negative instances. It only uses positive samples to adjust the model parameters in training. Figure 3.1(d) depicts the batch composition as described above.

3.2.2 Triplet loss and Its Hybrids with Positive Pair Loss

"Triplet loss" function extends "Large Margin Neighbour Loss" [65], which comprises of two terms, a pull-term and a push-term. The pull-term pulls data points i toward their corresponding target neighbors $T(i)$ of the same class. At the same time, the push-term enlarges the distance between points of different classes in the embedding space. The loss function defines as Eq. 3.6,

$$L_{LMNN} = (1 - \mu)L_{pull} + \mu L_{push} \tag{3.6}$$

$$L_{pull}(x_i, x_j) = \sum_{i,j \in T(i)} d(x_i, x_j) \tag{3.7}$$

$$L_{push}(x_i, x_j) = \sum_{i,j\&\ y_i \neq y_j} \{m + d(i, T(i)) - d(i, j)\}_+ \tag{3.8}$$

where the pull-loss (Eq. 3.7 [65]) calculates the distance between points and their expected target neighbors, and the push-loss (Eq. 3.8 [65]) accounts for the anchor-negative pairs that violate the constraints. However, the algorithm chooses the fixed target neighbors at the onset of training, making it less applicable in the scenario where target neighbors are changing dynamically in the training process. The "Triplet loss" addresses this issue by omitting the fixed target and choosing anchor (can be thought as the target neighbor) sample in run-time. Such design ensures that the anchor point a is closer to the positive point p than the negative point n in the projected embedding space. The formula below describes the above constraint (Eq. 3.9),

$$d(a,p) + m < d(a,n) \tag{3.9}$$

with a, n, p forming a triplet.

$$L_{Triplet} = \sum_{a,p,n \; y_a = y_p \neq y_n} \{m + d(a,p) - d(a,n)\}_+ \tag{3.10}$$

Consequently, it formulates a loss function (see Eq. 3.10 [63]), where $D(a,p)$ and $D(a,n)$ represent the distance between the anchor and the positive point and the distance between the negative and the anchor point in the embedding space, respectively. We make one intuitive illustration explaining the learning process (see Figure 3.2.(b)). Additionally, this loss only accounts for the triplets violating the constraint above when the anchor-negative distance is less than the sum of margin and anchor-positive distance. Similarly, in the paper [63], the authors define the hard-positive pair, hard-negative pair, as well as a more effective semi-hard negative pair. A hard-positive (Eq. 3.11 [63]) pair is formed by choosing a positive point from the batch with the maximum distance to anchor. In contrast, hard-negative (Eq. 3.12 [63]) pairs consist of anchors and negative samples with minimal distance. Likewise, semi-hard negative (Eq. 3.13 [63]) pairs have distance larger than that of anchor-positive pairs but smaller than the sum of the margin m and the distance of anchor-positive pair.

$$\arg\max_{x_i^p} d(x_i^a, x_i^p) \tag{3.11}$$

$$\arg\min_{x_i^n} d(x_i^a, x_i^n) \tag{3.12}$$

$$d(x_i^a, x_i^p) < d(x_i^a, x_i^n) < d(x_i^a, x_i^p) + m \tag{3.13}$$

In a mini-batch, P classes are randomly sampled, and each class comes with a sample size of K, eventually giving a batch size of $P \times K$. The authors claim that using all anchor-positive pairs in the batch promotes a stable model and faster convergence. Another finding is the use of the hardest negative (having minimal distance to the positive anchor in a batch) examples in practice traps the model in the local minima, resulting in a collapsed model. Therefore, the use of the semi-hard negative pair is proposed. So, for each anchor-positive pair, a semi-hard negative pair is chosen to form a triplet. The overall batch formation process ultimately yields $P \times K \times (K-1)$ triplets. We apply the same triplet formation process in our implementation.

We notice that triplet loss, similarly, considers the pull loss (obtained from the anchor-positive pair) and a push loss (calculated from the anchor-negative pair) during the training step. Therefore, we assume the standard "Triplet loss" would suffer the same problem as "Contrastive loss" does. To accommodate the issue, we propose several adjustments to the design of "Triplet loss", which include using various sampling strategies and ignoring the contribution of the negative class during training.

Triplet loss

Unlike "Contrastive loss" and "cross-entropy loss" that use a random-sampling strategy, the design of "Triplet loss" is incompatible with random-sampling. Each triplet requires data from two classes, but the random-sampling strategy arbitrarily samples the data from classes, potentially resulting in zero number of positive instances and breaking the construction of triplets.

Triplet loss with over-sampling. Another straightforward approach for highlighting the minority class's importance is to over-sample the minority class. With more attention to the minority class, two classes can, hence, pay equal attention to model building. The sampling ratio we adopted here is 1:N, which means we sample out one negative instance whenever N positive examples are sampled (see Figure 3.1(c)).

Triplet loss with class-balanced-sampling. We add the class-balanced-sampling strategy (see Figure 3.1(a)) into the standard triplet loss to balance the unequal contributions of classes in the training phase.

Triplet loss + Positive Pair Loss with Different Sampling Strategy

Triplet loss + Positive Pair loss [66]. The original paper on which the method is proposed aims at solving the person re-identification challenge. This challenge has two major problems; one is to identify the same person, and the other one is to identify a similar pose for the same person. This loss function proposed has two-fold considerations,

1. promotes the compactness of clusters representing pictures of the same person (each person represents a class),

2. clusters pictures with similar poses of the same identities.

Besides, they mention the positive pair loss can deal with the illumination effect in the picture background. Therefore, we attempt to use this loss function in the prevention of subtle variance that would result in a bigger dissimilarity of the instance from the signal class. Besides, we incorporate this approach with a class-balanced-sampling strategy.

"Triplet loss + Positive Pair loss with over-sampling". We introduce the over-sampling strategy to this method in the hope of alleviating unequal contributions. We sample the equal-sized examples to form an initial batch and then change the number of samples in the batch to meet the 1:N ratio. The model then uses it as the input batch (see Figure 3.1(c)). Additionally, the formula of this loss function remains unchanged.

Triplet loss + Positive Pair loss with Different Positive Anchor Consideration.

Apart from using different sampling strategies, we adopt the same principle used in "Positive Pair Positive Anchor loss", which excludes the impact of the negative class. Here we add this technique into our "Triplet loss + Positive Pair loss" and end up having three variants, which are

1. "Triplet Positive Anchor loss + Positive Pair loss" (Triplet PA loss + Positive Pair loss),

2. "Triplet loss + Positive Pair Positive Anchor loss" (Triplet loss + Positive Pair PA loss),

3. "Triplet Positive Anchor loss + Positive Pair Positive Anchor loss" (Triplet PA loss + Positive Pair PA loss).

Please note that we abbreviate "Positive Anchor" to "PA".

"Triplet PA loss + Positive Pair loss". In this loss function, we consider the triplets formed by using positive examples as anchors only. Nevertheless, the "Positive

pair loss" considers the gradient from both classes. We assume that because there are more triplets formed by using negative examples as anchors, the negative class will have more dominant effects in parameters' update. Thus, the model treats the negative and positive classes differently and margins out the effect brought from positive instances. To reflect the objective, this loss function is then modified to Eq. 3.14, where i and j form pairs (P) sampled from the same class.

$$L = \sum_{a,p \in D_{positive} \text{ and } n \in D_{negative}} \max(d(a,p) - d(a,n) + m, 0) + \sum_{i,j \in P} d(i,j) \tag{3.14}$$

"Triplet loss + Positive Pair PA loss". We have disabled the contribution of triplet loss from negative instances in the previous approach. In this method, we maintain the contribution of negative triplets. Instead, we ignore the loss from negative-pairs counted in the "Positive pair loss". It is designed to compare with our previous method to see how much effect the loss of negative-pairs count. Thus, we change the formula to Eq. 3.15, where i and j form pairs sampled from the positive class.

$$L = \sum_{a,p,n \in D} \max(d(a,p) - d(a,n) + m, 0) + \sum_{i,j \in P_{positive}} d(i,j) \tag{3.15}$$

"Triplet PA loss + Positive Pair PA loss". Our last attempt is to completely block the contribution of negative instances in the "Triplet loss + Positive Pair loss". It can be achieved by not selecting all the negative samples as anchors when constructing triplets and omitting the loss from negative instances. Such a process ensures the final loss to be the results coming from the positive instances only. It is consequently changed to Eq. 3.16, where i and j form pairs sampled from the positive class.

$$L = \sum_{a,p \in D_{positive} \text{ and } n \in D_{negative}} \max(d(a,p) - d(a,n) + m, 0) + \sum_{i,j \in P_{positive}} d(i,j) \tag{3.16}$$

3.3 Comparisons of Cross-Entropy Based Learning and Metric-based Learning

We summarize the differences of cross-entropy based learning and metric-based learning in the Table 3.1

Methods	Cross-entropy based classifiers	Metric-based classifiers
Sampling and outliers	Less sensitive to sampling method and noisy data	Rely heavily on sampling method and prone to outliers [61]
Embedding space	Separated classes but no class compactness and inter-class margin maximization [67]	Intra-class compactness and inter-class margin maximization
Batch size	Larger batch size leads to deterministic optimization and small batch size results in oscillations	Larger batch size reduces outlier interference and smaller batch sizes decrease model's robustness

Table 3.1: Comparisons of cross-entropy based learning and metric-based learning

Chapter 4

Experiments and Analysis

In this chapter, we perform experiments to answer proposed questions, like the impact of few important aspects, which include the training size, the pretrained parameters and the data imbalance. Section 4.1 describes the setup of the experiments such as model architecture and optimization. Section 4.2 describes how we design experiments and the dataset we used to simulate different degrees of class imbalance as well as the amount of training examples. Section 4.4 provides empirical results and discussions.

4.1 Setup

In this section, we introduce Residual Network Architecture (ResNet) [19] and changes we made in the model structure for the experiments. Also, we describe the hardware and software used for this project.

4.1.1 Model Architecture

Convolution Neural Network, as discussed before, increases computation efficiency and memory efficacy while showing compelling performance. However, from observations in H. Kaiming and S.Jian's work [68] and the original paper [19], the deeper networks consisting of stacked convolution blocks experience performance degradation. To address it, they incorporate Residual blocks into their models, and extensive experiments justify that the use of Residual blocks boosts the model performance by a large margin. Figure 4.1 shows the design of the Residual block with the skip-connection. Skip-connection implements the idea of identity mapping, which ensures that the output of stacked layers $F(x) + x$ has minimal difference to the input x. This idea comes from an assumption that instead of using a few stacked layers to asymptotically approximate the desired mapping $H(x)$ from input x to output M, these nonlinear stacked layers can be used to approximate the residual function, represented by $H(x) - x$. This reformulation assumes output M, and input x are interchangeable, given the residue value to be 0. Thus, the new formula of mapping function becomes $F(x) + x$, where $F(x) = H(x) - x$.

Residual Network Architecture (ResNet) [19] won first place in many tasks in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and Microsoft Common Object in Context (MS COCO) competitions in 2015.

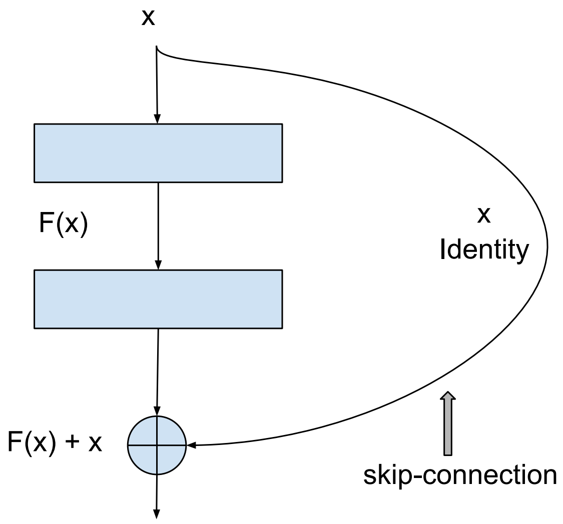


Figure 4.1: *Residual Learning Block*

Also, the authors propose several variants of Residual Network Architectures, such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152. We choose ResNet-18 as our model architecture because it has fewer parameters and is thus compatible with the size of our dataset.

The standard ResNet-18 model consists of one convolutional layer, one max pooling layer, four consecutive residual blocks, and a final fully connected layer. We replace the last fully connected layer of the ResNet-18 model with a linear layer with two output units because our goal is to learn a binary classification model. For the metric-based classifiers, the last layer is replaced with a linear layer with 512 output units, so that the model projects an input example to a hyperplane with 512 dimensions.

4.1.2 Software and Hardware

We conduct our experiments in Ubuntu 19.04 Desktop Version. It has 32 GB RAM and NVIDIA TITAN V GPU with 12 GB RAM. All the code is developed in PyCharm Community Version and managed by Git. The code is written in Python 3.6 and PyTorch 1.2.0.

4.2 Experiment Design

We design the experiments to answer the following questions:

I Q1: What is the impact of the transfer learning?

II Q2: What is the impact of the amount of training data?

III Q3: What is the impact of the imbalanced dataset? How much impact does it have?

IV Q4: How to choose between cross-entropy and metric-based methods?

To design experiments with respect to the above questions, we should consider two essential aspects:

- *Pretrained and Non-pretrained.* Transfer learning offers models with pretrained parameters (trained on ImageNet or other large data sources). In pretrained mode, we initialize the model with pretrained parameters from ImageNet. While in the non-pretrained setup, the model is randomly initialized from a normal distribution.

- *Balanced and Imbalanced.* We simulate the different number of training examples by taking a subset of the original dataset (see Section 2.2.1), with 16 examples per class as the smallest training set, to 1,204 per class as the largest dataset. And, we create different degrees of class imbalance, from extreme imbalance with 1,024 negative examples and 16 positive samples, to balanced data where both classes have 1,024 samples.

We group these conditions into four scenarios (see Table 4.1):

		Pretrained	
		NO	YES
Balanced	Yes	Balanced-Scratch	Balanced-Pretrained
	No	Imbalanced-Scratch	Imbalanced-Pretrained

Table 4.1: The breakdown of scenarios, with the options to use pretrained models displayed horizontally and the options to train models on a balanced dataset presented vertically.

1 Balanced training data with randomly initialized model parameters (Balanced-Scratch).

2 Balanced training data with pretrained model parameters from ImageNet (Balanced-Pretrained).

3 Imbalanced training data with randomly initialized model parameters (Imbalanced-Scratch).

4 Imbalanced training data with pretrained model parameters from ImageNet (Imbalanced-Pretrained).

We simulate different dataset compositions (in Table 4.2) so as to:

1 analyze model performance concerning the usage of pretrained parameters in different sizes of datasets (to answer Q1). We thereby compare model performance from Balanced-Pretrained to Balanced-Scratch and from Imbalanced-Pretrained to Imbalanced-Scratch, respectively ;

2 analyze the performance as we increase the amount of training data (to answer Q2). We measure the change of model performance with regard to the increase of data size in each scenario. ;

3 analyze model performance from a balanced dataset to an imbalanced version (to answer Q3). It is obtained by comparing model performance from Balanced-Scratch to Imbalanced-Scratch and Balanced-Pretrained to Imbalanced-Pretrained, separately.

Variances in initializing model parameters and also in sampling data are important factors that would lead to diverse model performance when other experimental factors remain the same. One way to measure these variances is to repeat the experiments; however the downside of this approach is the quadratic growth of training time. We use the same random seed 42 for model initialization and data sampling. Moreover, we set the optimization in the backend to be deterministic (features in Pytorch framework) to give reproducible results. In this way, we can ensure 1) models are initialized with the same weights and bias all the time; 2) the same training datasets are partitioned from the initial training; 3) The same sequence of data is sampled from the dataset in each iteration. In terms of the model optimization, we use Adam [69] optimizer and ReduceLROnPlateau for the optimization scheduler.

4.3 Evaluation and Baseline

A class-balanced hold-out test set is constructed for evaluation so as to ensure that the evaluation measure equally reflects the performances on both the minority class and the majority class. The test set is composed of 512 examples for each class. We use classification accuracy on the test set as the main evaluation measure. And, the precision and recall of the best methods are analyzed to compare with the baseline.

The main baseline for this thesis is the deep neural network proposed by Shiu et al. [1] for detecting the vocalizations of North Atlantic right whales. It is shown in their work that deep learning methods can achieve “false-positive rates that are orders of magnitude lower than alternative algorithms while substantially increasing the ability to detect calls” [1]. Their work also shows that deep neural networks can generalize a model trained in one geographical region to other regions and years.

Despite the success of deep learning in detecting the vocalizations, it is unclear what is the sample complexity and the sensitivity to class imbalance. We show in our experiments that, while achieving similar levels of precision and recall, the crossentopy and triplet loss based methods reduce the number of positive examples, i.e., upcalls, from around 6000 to 512, which indicates that the cross-entropy and triplet loss combined with specialized sampling can compensate for 90% less training data.

Experiments	Balanced Dataset (No.Neg, No.Pos)	Imbalanced Dataset (No.Neg, No.Pos)
Exp 1	(16,16)	(1024, 16)
Exp 2	(32,32)	(1024, 32)
Exp 3	(64,64)	(1024, 64)
Exp 4	(128,128)	(1024, 128)
Exp 5	(256,256)	(1024, 256)
Exp 6	(512,512)	(1024, 512)
Exp 7	(1024,1024)	(1024, 1024)

Table 4.2: The composition of balanced and imbalanced datasets. The first column displays various sizes of balanced datasets, with 16 positive samples and 16 negatives samples as the smallest and 1204 per class as the largest. The second column shows the composition of imbalanced datasets, which starts with 1024 negative samples and 16 positive samples and ends with 1024 per class.

4.4 Empirical Results

We first present the empirical results, which will be analyzed in Section 4.5. In this section, we analyze the empirical results of each scenario using the following procedure. We first evaluate different variants of the same model family and pick the best performing one as the most representative model for each family, i.e., triplet, contrastive, and cross-entropy loss. We then compare different families of methods to understand their performance differences under each scenario.

4.4.1 Balanced Data when Trained from Scratch

This scenario refers to the setup where the model parameters are randomly initialized—as opposed to initialization from another pretrained model in a transfer learning setup—and different classes have the same number of training examples, i.e., balanced data.

Main results. The performance of each model family—cross-entropy, triplet and contrastive loss—is visualized in Figure 4.2. “Triplet loss” works better than “cross-entropy loss” when the number of examples for each class is less than 256, and they have similar performances with access to more training data. This suggests the effectiveness of metric-based methods in the low data regime. However, “contrastive

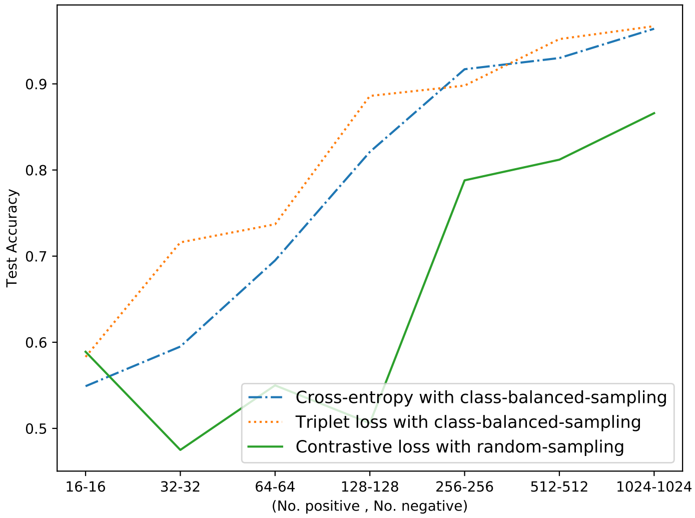


Figure 4.2: *Performance under Balanced Data when Trained from Scratch.*

loss" is inferior to other methods in most dataset compositions. This is because "contrastive loss" only considers the pair-wise relationship between data while ignoring more complex structures in the dataset. It is especially problematic in the PAM data because the distribution of the negative class, i.e., background, could have many different modes where different types of background noises interfere with pairwise contrastive training. However, this problem can be resolved by triplet loss because of the use of anchors; the triplet method aims to make the margin between positive examples smaller than the distances between positive-negative pairs with some margin.

More detailed empirical results and comparisons within each model family can be found in Appendix A.1.

4.4.2 Balanced Data with Transfer Learning

Section 4.4.1 discussed the results when models are trained from random initialization. However, transfer learning has become a prevalent approach in training deep learning models where we initialize a model's parameter from parameters from another model trained on large-scale datasets. In this section, we aim to investigate whether transfer parameters trained from ImageNet could benefit learning from ocean acoustic data.

It is worthwhile to note that the nature of tasks between ImageNet and ocean acoustic data are highly different, which raises the question of transferability between those datasets.

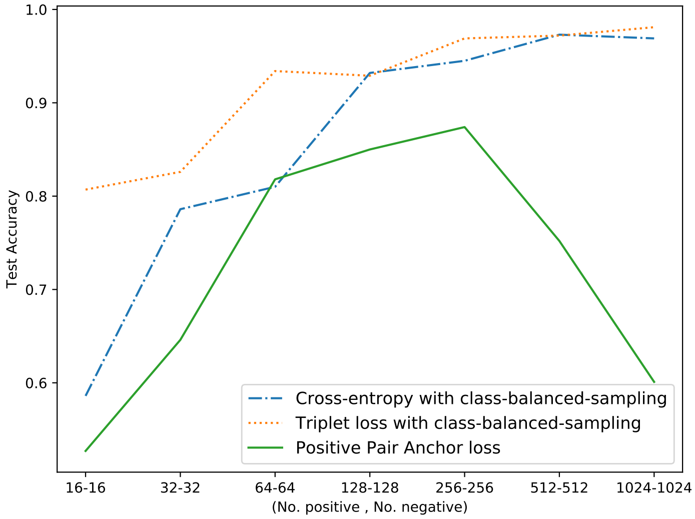


Figure 4.3: *Performance under Balanced Data with Pretrained Parameters.*

Main results. We show the test accuracy of representative classifiers from each model family in Figure 4.3. It is clear that "Triplet loss" is the best performing method among three classifiers due to its consist superior performance. It achieves 80% classification accuracy even with 16 examples per class—a 20% improvement from other methods. Meanwhile, "cross-entropy" is also quite stable and the generalization performance improves monotonically with the access of more labeled training data, and the gap between "triplet" and "cross entropy" diminishes with more training data. Similar to findings in Section 4.4.1, "contrastive loss" does not work well.

The impact of transfer learning. Figure 4.4 compares the performance with and without transfer learning when the datasets are balanced. Surprisingly, we find that, although ImageNet and spectrogram obtained from ocean acoustic data are from different domains, the use of pretrained parameters from ImageNet is highly beneficial to the generalization performance. The improvements are more profound when the amount of labeled training data is small, which can better leverage the inductive bias learned from the large-scale ImageNet dataset.

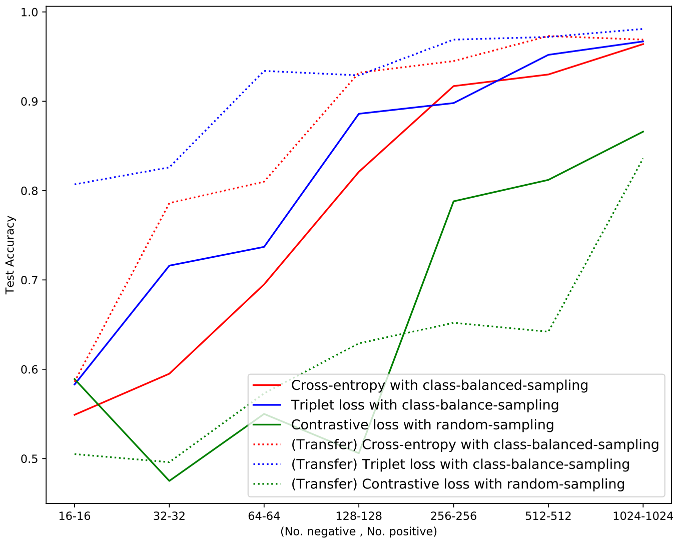


Figure 4.4: *The impact of transfer learning on balanced datasets*

More detailed empirical results and comparisons within each model family can be found in Appendix A.2.

4.4.3 Imbalanced Data when Trained from Scratch

We conduct this set of experiments on simulated imbalanced datasets when the models are trained from random initialization.

The impact of over-sampling. We show in Figure 4.5 that "cross-entropy" is sensitive to the sampling strategy and works the best when the classes of each batch are balanced. Moreover, the random sampling approach works the worst which suggests the need for sampling under class imbalance. Similar findings are found in Figure 4.6 for "triplet loss". We conclude that balanced sampling is necessary given dataset imbalance.

Comparison of different model families. Figure 4.7 shows that "triplet loss" tends to work better than "cross-entropy" and "contrastive loss". And, we observe that "cross-entropy with class-balance-sampling" classifier outperforms Triplet loss classifiers only in one experiment and remains a slight disadvantage throughout most experiments. "Contrastive loss with random-sampling" classifier performs the worst in all experiments.

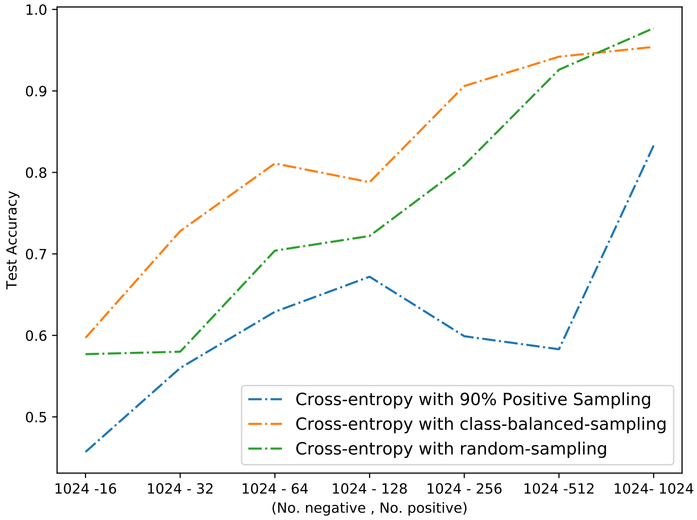


Figure 4.5: *Performance of cross-entropy based classifiers in Imbalanced-Scratch Scenario.*

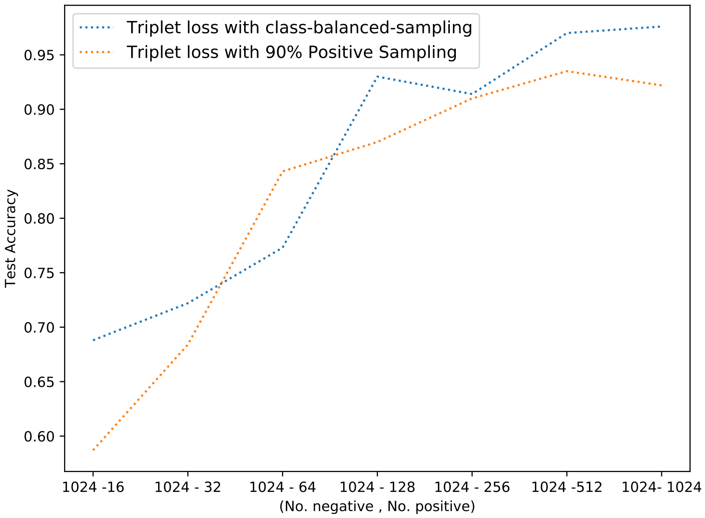


Figure 4.6: *Performance of Triplet loss based classifiers in Balanced-Pretrained Scenario.*

More detailed empirical results and comparisons within each model family can be found in Appendix A.3.

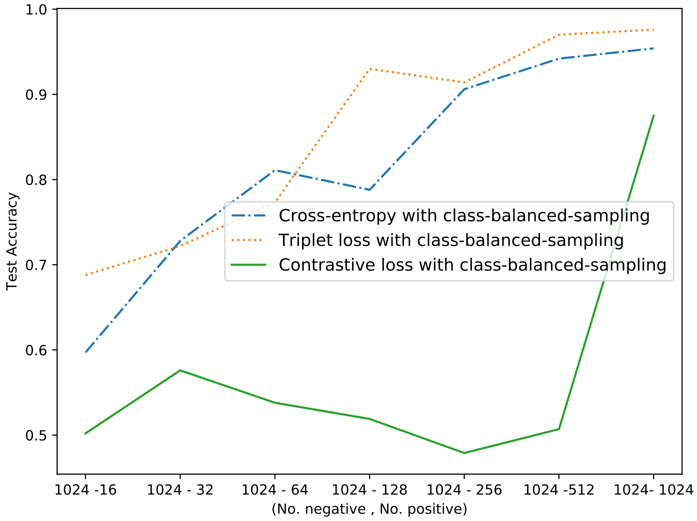


Figure 4.7: *Performance under Imbalanced Data when Trained from Scratch.*

4.4.4 Imbalanced Data with Transfer Learning

Section 4.4.3 discussed the results when models are trained from random initialization. In this section, we aim to investigate whether transfer parameters trained from ImageNet could benefit learning from imbalanced ocean acoustic data.

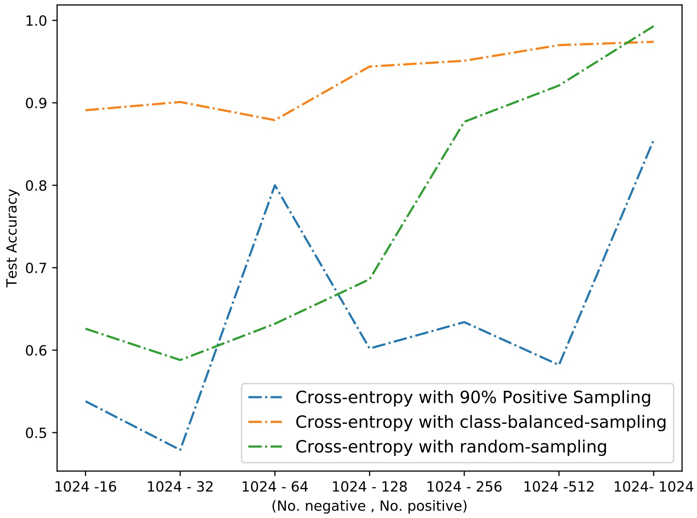


Figure 4.8: *Performance of cross-entropy based classifiers in Imbalanced-Pretrained Scenario.*

The impact of over-sampling. We have similar findings with Section 4.4.3 that "cross-entropy" is sensitive to the sampling strategy and works the best when the classes of each batch are balanced, as shown in Figure 4.8. Moreover, the random sampling approach works the worst which suggests the need for sampling under class imbalance. Similar findings are found for "triplet loss" in Figure 4.9. We argue that the oscillation of "Triplet loss with 90% Positive Sampling“ in smaller datasets results from small sample complexity of negative instances. The over-sampling strategy might introduce noisy data from limited sampled negative instances, and then these noisy data would influence model's stability.

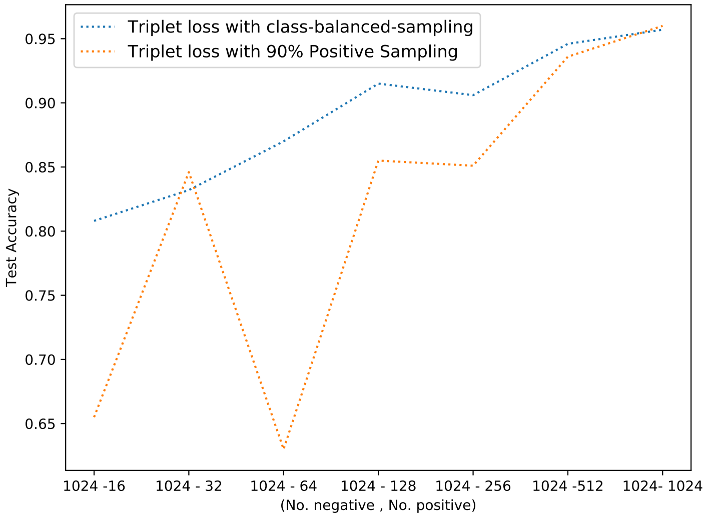


Figure 4.9: *Comparison of performance of Triplet loss based classifier in Imbalanced-Pretrained Scenario.*

Comparison between different model families. The performance of different model families are shown in Figure 4.10. It is clear that "cross-entropy with balance-sampling" outperforms other approaches. Both "cross-entropy with balance-sampling" and "Triplet loss" remarkably outperform "contrastive loss". One reason that "cross-entropy" outperforms the "triplet loss" under the transfer learning setup is that the model is trained using "cross-entropy" on ImageNet, which is more consistent with "cross-entropy" based fine-tuning on the ocean acoustic data. Therefore, the current comparison is biased towards the pretraining strategy. We consider using metric loss to pretrain the ImageNet in future work to further evaluate the impact of transfer

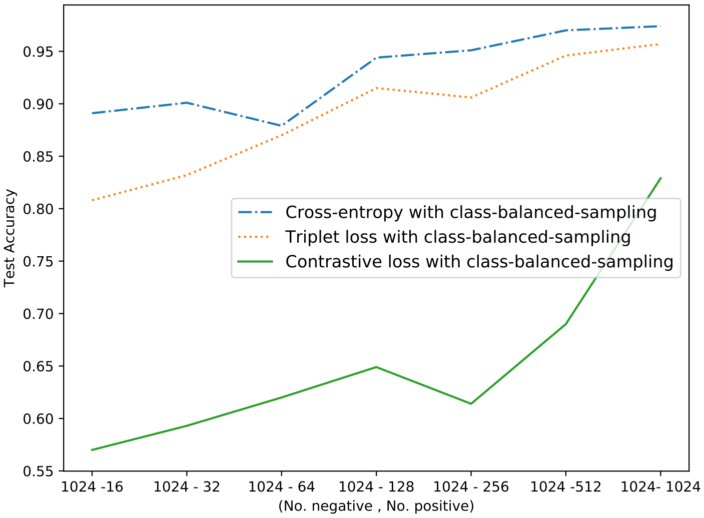


Figure 4.10: *Performance under Imbalanced Data with Transfer Learning.*

learning.

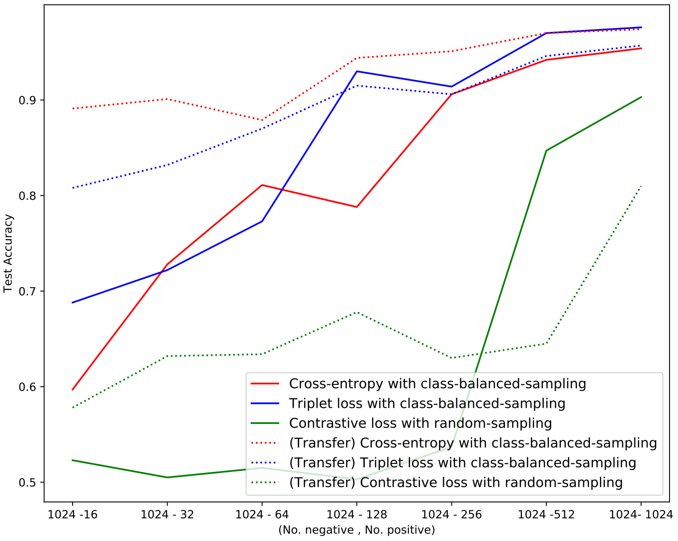


Figure 4.11: *The impact of transfer learning on imbalanced datasets.*

The impact of transfer learning. Figure 4.11 compares the performance with and without transfer learning when the datasets are imbalanced. Similar to Section 4.4.2, we find that, although ImageNet and spectrogram obtained from ocean acoustic data

	Scratch		Transfer	
dataset	cross-entropy with class-balanced-sampling	Triplet loss with class-balanced-sampling	cross-entropy with class-balanced-sampling	Triplet loss with class-balanced-sampling
1024-16	0.4978	0.1812	0.8176	0.7972
1024-32	0.6539	0.4874	0.8309	0.7766
1024-64	0.6909	0.7659	0.875	0.9007
1024-128	0.8691	0.8991	0.9003	0.8216
1024-256	0.8798	0.9036	0.9627	0.8948
1024-512	0.9336	0.9136	0.9566	0.9262
1024-1024	0.9802	0.9618	0.9722	0.975

Table 4.3: Comparison of F1-score for methods trained with and without transfer learning.

are from different domains, the use of pretrained parameters from ImageNet is highly beneficial to the generalization performance. The improvements are more profound when the amount of labeled training data is small, which can better leverage the inductive bias learned from the large-scale ImageNet dataset.

More detailed empirical results and comparisons within each model family can be found in Appendix A.4.

4.5 Analysis

In this sections, we aim to answer the research questions proposed in Section 4.2.

4.5.1 The Impact of Transfer Learning

Table 4.3 compares the F1-scores between models trained with and without transfer learning. We find that transfer learning from ImageNet provides substantial improvements in the F1-score of both cross-entropy and triplet based methods. The use of transfer learning is especially important in the low-shot end of the spectrum where only 16 positive examples are available: transfer learning improves the f1-score from 49% to 81% for cross-entropy based methods and from 18% to 79% for metric based methods.

Table 4.4 shows the comparison of Average Precision (AP) between models trained with and without transfer learning. The formula of AP is defined below (see Eq: 4.1).“AP summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the

Average Precision				
	Scratch		Transfer	
Dataset	Cross-entropy with class-balanced-sampling	Triplet loss with class-balanced-sampling	Cross-entropy with class-balanced-sampling	Triplet loss with class-balanced-sampling
1024-128	0.9342	0.9292	0.9815	0.8866
1024-256	0.9697	0.9017	0.9966	0.9266
1024-512	0.9933	0.9725	0.9952	0.9482
1024-1024	0.9964	0.9403	0.9978	0.9775

Table 4.4: Comparison of Average Precision (AP) for methods trained with and without transfer learning.

weights

$$AP = \sum_{n}(R_n - R_{n-1})P_n \tag{4.1}$$

, where P_n and R_n are the precision and recall at the n_{th} threshold." [70]. From Table 4.4, we observe that transfer learning indeed helps improve the average precision for "cross-entropy with class-balanced-sampling" but this is less obvious for the "Triplet loss with class-balanced-sampling." Moreover, the increase of average precision becomes minimal between two relatively larger datasets (1024-256 and 1024-512) for "cross-entropy with class-balanced-sampling" after using transfer learning. Moreover, the advantage of using transfer learning decreases with increased data. For example, there is a 4.7% improvement in the 1024-128 dataset for "cross-entropy with class-balanced-sampling" after using the transfer learning, but the improvement reduces to 0.2% in the 1024-512 dataset. What's more, transfer learning does not show consistent improvement for "Triplet loss with class-balanced-sampling". For example, the average precision in the 1024-128 dataset drops from 92.92% to 88.66% after using transfer learning. Similarly, another decrease is found in the 1024-512 dataset. We visualize the precision-recall curves for different datasets in Figure 4.12, Figure 4.13, Figure 4.14 and Figure 4.15. By looking at these precision-recall curves, we find that "cross-entropy with class-balanced-sampling" with transfer learning (in green) performs better (having larger area under curve) than all other methods in the 1024-128 and 1024-256 datasets, and shows comparable performance to "Triplet loss with class-balanced-sampling" in other settings.

In conclusion, we find that transfer learning helps improve model performance in terms of accuracy, F1-score and average precision, especially for "cross-entropy" based classifiers, but provides a less observable boost for “Triplet loss” based classifiers. Also, the benefit of using transfer learning decreases as more data are available. Moreover, our classifiers, which are trained with less and imbalanced data, demonstrate comparable performance to baseline classifiers with average precisions of 0.903 and 0.891, respectively [1].

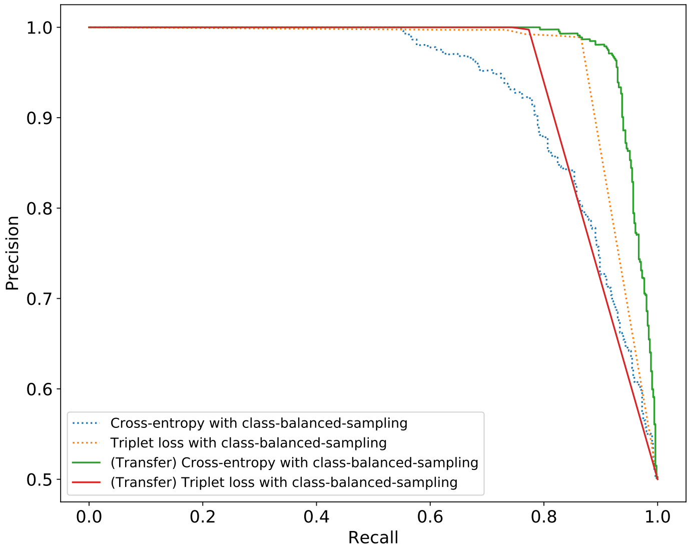


Figure 4.12: *Precision-Recall curve for methods (trained with and without transfer learning) on the 1024-128 dataset*

4.5.2 The Impact of Dataset Imbalance

The empirical results suggest that class imbalance indeed impairs the performance of classification methods.

Both “cross-entropy with class-balanced-sampling” with pretrained parameters and Triplet loss based classifiers tend to perform well in the presence of class imbalance. Empirical results from section 4.4.1 to section 4.4.2 suggest that Triplet loss based classifiers can better handle class imbalance than most methods, except the “cross-entropy with class-balanced-sampling” using pretrained parameters.

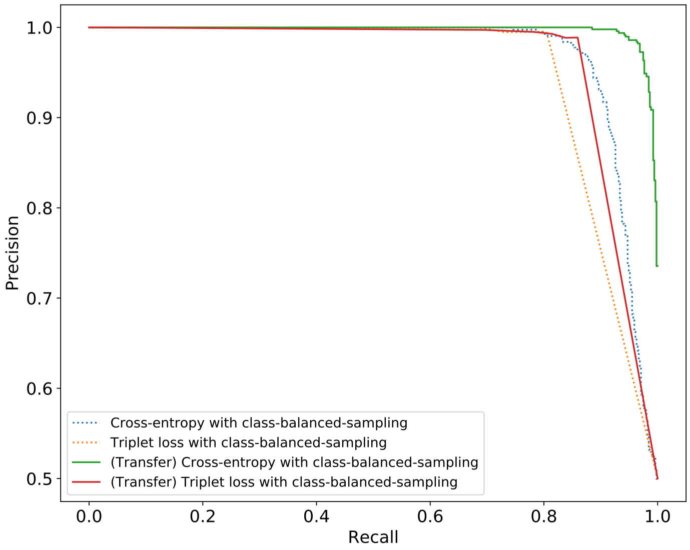


Figure 4.13: *Precision-Recall curve for methods (trained with and without transfer learning) on the 1024-256 dataset*

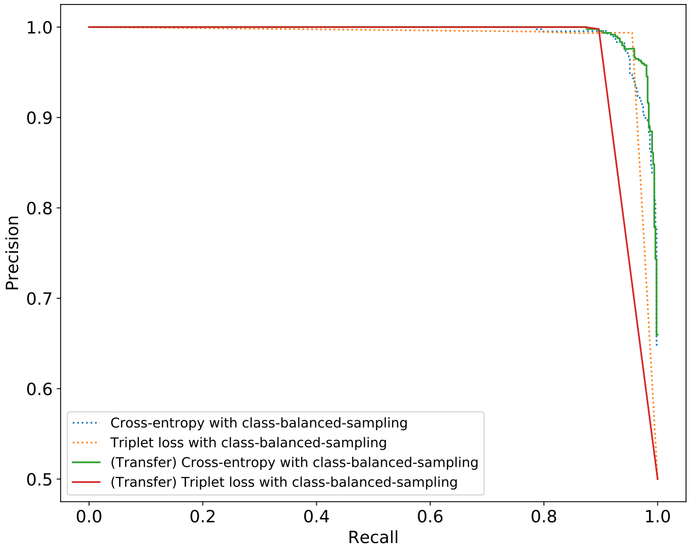


Figure 4.14: *Precision-Recall curve for methods (trained with and without transfer learning) on the 1024-512 dataset*

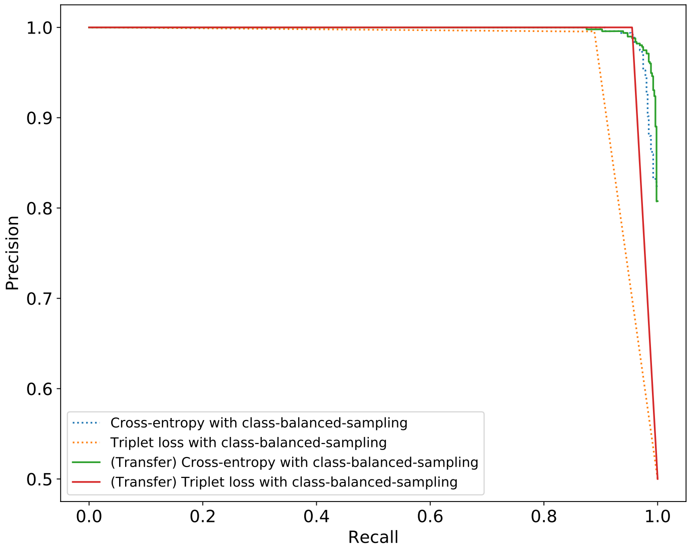


Figure 4.15: *Precision-Recall curve for methods (trained with and without transfer learning) on the 1024-1024 dataset.*

4.5.3 The Impact of The Amount of Training Data

In most experiments, we have observed that when increasing the training data, models tend to have higher accuracy, which further indicates that performance growth and increasing training data size are positively correlated. It is a general phenomenon in machine learning due to the Probably Approximately Correct (PAC) framework [71], which says that with the same model, the more samples used for training, the smaller generalization error the model will get.

The quantity of training data profoundly impacts the model's performance, as more training data promotes better model performance. It is observed that the model behaves differently to the increased data. Through observation, we notice that two factors are contributing to the slower model growth, which are higher initial model performance and inherent design failures. The higher initial model performance leaves the model less room for vast improvement. The other problem exists in the design of the loss function, for instance, "Positive Pair Anchor loss". It focuses on the positive sample only, and this design results in low predictability for the negative class. Therefore, even when data size increases, the model's capability is constrained by the negative predictability so that it saturates at 50% (maximum predictability for the positive class). This problem also appears in some variants of contrastive loss based classifiers when the number of negative instances in the batch is smaller than that of positive examples.

4.6 Summary

Through experiments and analysis, we have the following findings:

1 From Section 4.5.1, we observe that the transfer learning helps cross-entropy based classifier combined with class-balanced-sampling outperform all studied methods by a clear margin in the heavily imbalanced dataset. The Triplet loss with transfer learning displays comparable performance to "cross-entropy with class-balanced-sampling" in most cases. Moreover, the non-pretrained classifiers trained with Triplet loss and its hybrids surpass the classifiers trained with cross-entropy loss in many scenarios. Contrastive loss is generally ineffective, regardless of whether pretrained parameters are used. The pretrained parameters are better suited for the classifiers with cross-entropy losses than the models using metric-based losses because the pretraining is based on cross-entropy.

2 From Section 4.5.2, we find that the imbalance datasets degrade classifiers' performance in most cases. But from a series of analysis, Triplet based classifiers are shown to be able to better handle the shift from balanced datasets to imbalanced datasets and to maintain reasonable performance, with or without the use of pretrained parameters. Cross-entropy based classifiers, in general, are less capable of handling the imbalance. Contrastive based classifiers, except variants of Positive Pair loss, display positive increases in datasets shifts.

3 From Section 4.5.3, we find that the training data size influences the cross-entropy based classifiers and metric-based classifiers differently. In the balanced scenarios, cross-entropy based classifiers in general have unsatisfactory results with insufficient data while the metric-based classifiers (Triplet based classifiers) still perform well in small datasets. Even in small-sized imbalanced scenarios, Triplet based classifiers demonstrate comparable or better performance compared to cross-entropy based classifiers. And, a slight increase in the dataset size tends to strongly influence the performance of cross-entropy loss and Triplet

loss based classifiers (improving their performance hugely), but not Contrastive loss based classifiers. Moreover, most classifiers tend to work better with larger datasets.

4 From Section 4.4.1 to Section 4.4.4, we find the degree of oversampling matters. Extreme oversampling with the positive classes leads to degraded performances because the model fails to learn adequate information from the negative examples.

We conclude the experiment section with the following summaries:

1 The pretrained settings and class-balanced-sampling offer the model the ability to handle a heavily imbalanced dataset.

2 The pretrained parameters, on the one hand, help the cross-entropy based classifiers achieve impressive results but, on the other hand, result in a relatively less compelling improvement for metric-based models.

3 The advantage of using pretrained parameters decreases as the training data increases. As long as a larger volume of the training dataset is available, models can obtain comparable performance to the models that use pretrained parameters and train on small datasets.

4 For effective training on the small imbalanced datasets, we recommend training a pretrained cross-entropy based classifier and using a class-balanced-sampling strategy. If a pretrained model is unavailable, the safest alternative is to train a Triplet loss based classifier. As for balanced datasets, Triplet loss based classifiers tend to be better options.

Chapter 5

Conclusion and Future Work

In this thesis, we describe our motivation for applying deep learning on the underwater acoustic data classification task and briefly discuss existing works that have used machine learning techniques to alleviate human effort. We then introduce the data collection process via Passive Acoustic Monitoring (PAM) and emphasize the importance of converting the raw audio data into a spectrogram, which is a two-dimensional representation reflecting the correlation between time and frequency. Our research of interest is to compare two mainstream types of classifiers, cross-entropy based and metric-based, on acoustic data. We train these classifiers with various training loss functions to reflect particular focuses on different aspects of the classification task. We then evaluate those classifiers on four different scenarios based on whether the data is imbalanced and whether pretrained model parameters are available: Balanced-Scratch, Balanced-Pretrained, Imbalanced-Scratch, and Imbalanced-Pretrained. Experimental results are analyzed to shed light on the strengths and weaknesses of each method and the importance of the pretrained parameters, data imbalance, as well as the size of training data. We conclude that

1 Pretrained parameters enable the model to achieve higher performance when minimal data are available, but they add a less impressive boost to the classes when more data are presented.

2 Imbalanced data impairs the model performance by decreasing predictability for the minority class, but some classifiers with special adjustments (re-sampling and pretrained settings) are immune to the imbalance.

3 The size of the training dataset strongly influences the model performance; larger size promotes higher performance while a smaller amount of examples leads to inferior model performance.

4 Metric-based classifiers, typically Triplet loss based classifiers, work well in most

situations while cross-entropy based classifiers perform the best only in the imbalanced dataset with pretrained parameters and class-balanced-sampling.

However, our work has the following limitations,

1 the task we deal with in this thesis is a binary classification, and the implementation does not work with the multi-label classifications,

2 we only explored a small subset of the DCLDE 2013 dataset, and datasets are obtained over a single week in Massachusetts Bay. Therefore, these data reflect only a small sample of marine animals and environmental conditions, which lack generalizability to the entire population of NARWs [1],

3 the experiments are conducted with constant random seed (42) to give reproducible results in author's machine; however, the same results are difficult to obtain in different machines. Variance in initialized model parameters and data sampling would give diverse model performance. On the one hand, the repeated experiments with different random seeds could help us measure the model variance and estimate the averaged model performance; on the other hand, it would multiply the computational time and increase the research cost.

As for future work, we first plan to extend our method to work with multi-label classification because when working with many classes, the class-separable metric space is the key to the better model performance. Moreover, with more classes presented in the dataset, the few-shot learning is another interesting direction to investigate. The second direction is to pretrain a Triplet model on richer acoustic datasets and to use it to initialize our triplet model to classify underwater acoustic data. It is because our current Triplet model with pretrained parameters is biased towards the pretraining strategy, i.e., cross-entropy on ImageNet, it hence does not perform well on current pretrained experiments. In the experiments, we found that the upcall has a unique pattern (a diagonally upward curve); therefore, the third direction is to design a specific kernel (filter for CNN) to detect this pattern and to see whether the customized kernel could improve the model performance. The fourth direction is to augment the NARW dataset. The naive approach is to shift the signal along time-axis to produce more positive samples. Others include SampleRNN [72], SpecAugment [73], Wavenet [74] and others [75], [76].

Further, we will investigate image retrieval tasks in the underwater acoustic domain so as to retrieve spectrogram of interest from large volumes of unlabeled data.

References

[1] Yu Shiu, KJ Palmer, Marie A Roch, et al. "Deep neural networks for automated detection of marine mammal species". In: *Scientific Reports* 10.1 (2020), pp. 1–12.

[2] Frans van Bommel BirdLife. *Birds in Europe: population estimates, trends and conservation status.* Cambridge, 2004.

[3] Samuel A Cushman. "Effects of habitat loss and fragmentation on amphibians: a review and prospectus". In: *Biological conservation* 128.2 (2006), pp. 231–240.

[4] Bradford Nick. *Marine Species on the Move.* en. URL: `https://www.neefusa.org/weather-and-climate/marine-species-move` (visited on 02/19/2020).

[5] Cédric Gervaise, Yvan Simard, Florian Aulanier, et al. *Optimal passive acoustic systems for real-time detection and localization of North Atlantic right whales in their feeding ground off Gaspé in the Gulf of St. Lawrence.* Department of Fisheries and Oceans, 2019.

[6] Tiago A Marques, Lisa Munger, Len Thomas, et al. "Estimating North Pacific right whale Eubalaena japonica density using passive acoustic cue counting". In: *Endangered Species Research* 13.3 (2011), pp. 163–172.

[7] Enrico Pirotta, Paul M Thompson, Peter I Miller, et al. "Scale-dependent foraging ecology of a marine top predator modelled using passive acoustic data". In: *Functional ecology* 28.1 (2014), pp. 206–217.

[8] Susanna B Blackwell, Christopher S Nations, Trent L McDonald, et al. "Effects of airgun sounds on bowhead whale calling rates: evidence for two behavioral thresholds". In: *PloS one* 10.6 (2015).

[9] John A Hildebrand, Simone Baumann-Pickering, Kaitlin E Frasier, et al. "Passive acoustic monitoring of beaked whale densities in the Gulf of Mexico". In: *Scientific reports* 5 (2015), p. 16343.

[10] Dan Cireşan, Ueli Meier, Jonathan Masci, et al. “A committee of neural networks for traffic sign classification”. In: *The 2011 international joint conference on neural networks*. IEEE, 2011, pp. 1918–1921.

[11] Douglas Gillespie. “Detection and classification of right whale calls using an’edge’detector operating on a smoothed spectrogram”. In: *Canadian Acoustics* 32.2 (2004), pp. 39–47.

[12] Oliver S Kirsebom, Fabio Frazao, Yvan Simard, et al. “Performance of a deep neural network at detecting North Atlantic right whale upcalls”. In: *The Journal of the Acoustical Society of America* 147.4 (2020). Publisher: Acoustical Society of America, pp. 2636–2646.

[13] Seppo Fagerlund. “Bird species recognition using support vector machines”. In: *EURASIP Journal on Advances in Signal Processing* 2007.1 (2007), p. 038637.

[14] Julie N Oswald, Jay Barlow, and Thomas F Norris. “Acoustic identification of nine delphinid species in the eastern tropical Pacific Ocean”. In: *Marine mammal science* 19.1 (2003), pp. 20–037.

[15] Thomas Guilment, Francois-Xavier Socheleau, Dominique Pastor, et al. “Sparse representation-based classification of mysticete calls”. In: *The Journal of the Acoustical Society of America* 144.3 (2018), pp. 1550–1563.

[16] Xanadu C Halkias, Sébastien Paris, and Hervé Glotin. “Classification of mysticete sounds using machine learning techniques”. In: *The Journal of the Acoustical Society of America* 134.5 (2013), pp. 3496–3505.

[17] Ya-Jie Zhang, Jun-Feng Huang, Neng Gong, et al. “Automatic detection and classification of marmoset vocalizations using deep and recurrent neural networks”. In: *The Journal of the Acoustical Society of America* 144.1 (2018), pp. 478–487.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, et al. "Deep Residual Learning for Image Recognition". In: *arXiv:1512.03385 [cs]* (Dec. 2015). arXiv: 1512.03385. URL: `http://arxiv.org/abs/1512.03385` (visited on 02/19/2020).

[20] Olga Russakovsky, Jia Deng, Hao Su, et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[21] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[23] Mahmut Kaya and H.s Bilge. "Deep Metric Learning: A Survey". In: *Symmetry* 11 (2019), p. 1066. DOI: `10.3390/sym11091066`.

[24] *Ocean Networks Canada*. en. Research. URL: `https://www.oceannetworks.ca/` (visited on 03/27/2020).

[25] David K Mellinger, Kathleen M Stafford, Sue E Moore, et al. "An overview of fixed passive acoustic observation methods for cetaceans". In: *Oceanography* 20.4 (2007), pp. 36–45.

[26] David K Mellinger. "A comparison of methods for detecting right whale calls". In: *Canadian Acoustics* 32.2 (2004), pp. 55–65.

[27] Janice M Waite, Kate Wynne, and David K Mellinger. "Documented sighting of a North Pacific right whale in the Gulf of Alaska and post-sighting acoustic monitoring". In: *Northwestern Naturalist* 84.1 (2003). Publisher: JSTOR, pp. 38–43.

[28] Lisa M Munger, David K Mellinger, Sean M Wiggins, et al. "Performance of spectrogram cross-correlation in detecting right whale calls in long-term recordings from the Bering Sea". In: *Canadian Acoustics* 33.2 (2005), pp. 25–34.

[29] Mark F Baumgartner and Sarah E Mussoline. "A generalized baleen whale call detection and classification system". In: *The Journal of the Acoustical Society of America* 129.5 (2011), pp. 2889–2902.

[30] Yann LeCun, Léon Bottou, Yoshua Bengio, et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[31] Jorge Beltran, Carlos Guindel, Francisco Miguel Moreno, et al. "Birdnet: a 3d object detection framework from lidar information". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3517–3523.

[32] Mahdi Esfahanian, Hanqi Zhuang, Nurgun Erdol, et al. "Detection of north atlantic right whale upcalls using local binary patterns in a two-stage strategy". In: *arXiv preprint arXiv:1611.04947* (2016).

[33] Ildar R Urazghildiiev, Christopher W Clark, Timothy P Krein, et al. "Detection and recognition of North Atlantic right whale contact calls in the presence of ambient noise". In: *IEEE Journal of Oceanic Engineering* 34.3 (2009), pp. 358–368.

[34] Stephanie Taylor and Tony R. Walker. "North Atlantic right whales in danger". en. In: *Science* 358.6364 (Nov. 2017), pp. 730–731. ISSN: 0036-8075, 1095-9203. DOI: `10.1126/science.aar2402`. URL: `https://science.sciencemag.org/content/358/6364/730.2` (visited on 02/20/2020).

[35] Christopher W Clark, Douglas Gillespie, Douglas P Nowacek, et al. "Listening to their world: acoustics for monitoring and protecting right whales in an urbanized ocean". In: *The urban whale: North Atlantic right whales at the crossroads. Harvard University Press, Cambridge, MA* (2007), pp. 333–357.

[36] Frazao Fabio and Kirsebom Oliver. *Ketos*. Dalhousie University, Apr. 2019. URL: `https://docs.meridian.cs.dal.ca/ketos/` (visited on 04/16/2020).

[37] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

[38] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[39] Pedro Domingos. “A few useful things to know about machine learning”. In: *Communications of the ACM* 55.10 (2012). Publisher: ACM New York, NY, USA, pp. 78–87.

[40] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. “Special issue on learning from imbalanced data sets”. In: *ACM SIGKDD explorations newsletter* 6.1 (2004). Publisher: ACM New York, NY, USA, pp. 1–6.

[41] Chris Drummond, Robert C Holte, et al. “C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling”. In: *Workshop on learning from imbalanced datasets II*. Vol. 11. Citeseer, 2003, pp. 1–8.

[42] Charles Elkan. “The foundations of cost-sensitive learning”. In: *International joint conference on artificial intelligence*. Vol. 17. Issue: 1. Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.

[43] Justin M Johnson and Taghi M Khoshgoftaar. “Survey on deep learning with class imbalance”. In: *Journal of Big Data* 6.1 (2019). Publisher: Springer, p. 27.

[44] Harsurinder Kaur, Husanbir Singh Pannu, and Avleen Kaur Malhi. “A systematic review on imbalanced data challenges in machine learning: Applications and solutions”. In: *ACM Computing Surveys (CSUR)* 52.4 (2019). Publisher: ACM New York, NY, USA, pp. 1–36.

[45] Miroslav Kubat, Stan Matwin, et al. “Addressing the curse of imbalanced training sets: one-sided selection”. In: *Icml*. Vol. 97. Nashville, USA, 1997, pp. 179–186.

[46] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. “Exploratory undersampling for class-imbalance learning”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.2 (2008). Publisher: IEEE, pp. 539–550.

[47] Inderjeet Mani and I Zhang. “kNN approach to unbalanced data distributions: a case study involving information extraction”. In: *Proceedings of workshop on learning from imbalanced datasets*. Vol. 126. 2003.

[48] Maxime Oquab, Leon Bottou, Ivan Laptev, et al. “Learning and transferring mid-level image representations using convolutional neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2014, pp. 1717–1724.

[49] Gary M Weiss. “Mining with rarity: a unifying framework”. In: *ACM Sigkdd Explorations Newsletter* 6.1 (2004). Publisher: ACM New York, NY, USA, pp. 7–19.

[50] Zhi-Hua Zhou and Xu-Ying Liu. “Training cost-sensitive neural networks with methods addressing the class imbalance problem”. In: *IEEE Transactions on knowledge and data engineering* 18.1 (2005). Publisher: IEEE, pp. 63–77.

[51] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, et al. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[52] Hengshun Zhou, Xue Bai, and Jun Du. “An investigation of transfer learning mechanism for acoustic scene classification”. In: *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP).* IEEE, 2018, pp. 404–408.

[53] Prerna Arora and Reinhold Haeb-Umbach. “A study on transfer learning for acoustic event detection in a real life scenario”. In: *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP).* IEEE, 2017, pp. 1–6.

[54] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, et al. “A closer look at few-shot classification”. In: *arXiv preprint arXiv:1904.04232* (2019).

[55] Szu-Yu Chou, Kai-Hsiang Cheng, Jyh-Shing Roger Jang, et al. “Learning to match transient sound events using attentional similarity for few-shot sound recognition”. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2019, pp. 26–30.

[56] Xiang Jiang, Liqiang Ding, Mohammad Havaei, et al. “Task Adaptive Metric Space for Medium-Shot Medical Image Classification”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2019, pp. 147–155.

[57] Kazuki Shimada, Yuichiro Koyama, and Akira Inoue. “Metric Learning with Background Noise Class for Few-Shot Detection of Rare Sound Events”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 616–620.

[58] Jake Snell, Kevin Swersky, and Richard Zemel. “Prototypical networks for few-shot learning”. In: *Advances in neural information processing systems*. 2017, pp. 4077–4087.

[59] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, et al. “Matching networks for one shot learning”. In: *Advances in neural information processing systems*. 2016, pp. 3630–3638.

[60] Yu Wang, Justin Salamon, Nicholas J Bryan, et al. “Few-Shot Sound Event Detection”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 81–85.

[61] Yao Zhai, Xun Guo, Yan Lu, et al. “In defense of the classification loss for person re-identification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 0–0.

[62] Raia Hadsell, Sumit Chopra, and Yann LeCun. “Dimensionality reduction by learning an invariant mapping”. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*. Vol. 2. IEEE, 2006, pp. 1735–1742.

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

[64] Dumitru Erhan, Yoshua Bengio, Aaron Courville, et al. “Why does unsupervised pre-training help deep learning?” In: *Journal of Machine Learning Research* 11.Feb (2010), pp. 625–660.

[65] Kilian Q Weinberger and Lawrence K Saul. “Distance metric learning for large margin nearest neighbor classification”. In: *Journal of Machine Learning Research* 10.Feb (2009), pp. 207–244.

[66] Paul Wohlhart and Vincent Lepetit. "Learning descriptors for object recognition and 3d pose estimation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3109–3118.

[67] Ahmed Taha, Yi-Ting Chen, Teruhisa Misu, et al. "Boosting Standard Classification Architectures Through a Ranking Regularizer". In: *arXiv e-prints* (Jan. 2019), arXiv:1901.08616.

[68] Kaiming He and Jian Sun. "Convolutional neural networks at constrained time cost". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 5353–5360.

[69] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[70] *sklearn.metrics.average_precision_score — scikit-learn 0.22.2 documentation*. en. Publication Title: scikit-learn Type: software. 2007. URL: `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html#rcdf8f32d7f9d-1` (visited on 04/05/2020).

[71] Leslie G Valiant. "A theory of the learnable". In: *Communications of the ACM* 27.11 (1984). Publisher: ACM New York, NY, USA, pp. 1134–1142.

[72] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, et al. "SampleRNN: An unconditional end-to-end neural audio generation model". In: *arXiv preprint arXiv:1612.07837* (2016).

[73] Daniel S Park, William Chan, Yu Zhang, et al. "Specaugment: A simple data augmentation method for automatic speech recognition". In: *arXiv preprint arXiv:1904.08779* (2019).

[74] Aaron van den Oord, Sander Dieleman, Heiga Zen, et al. "Wavenet: A generative model for raw audio". In: *arXiv preprint arXiv:1609.03499* (2016).

[75] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". In: *Journal of Big Data* 6.1 (2019). Publisher: Springer, p. 60.

[76] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, et al. "mixup: Beyond empirical risk minimization". In: *arXiv preprint arXiv:1710.09412* (2017).

Appendix A

Additional Empirical Results

A.1 Empirical Results with Balanced Data when Trained from Scratch

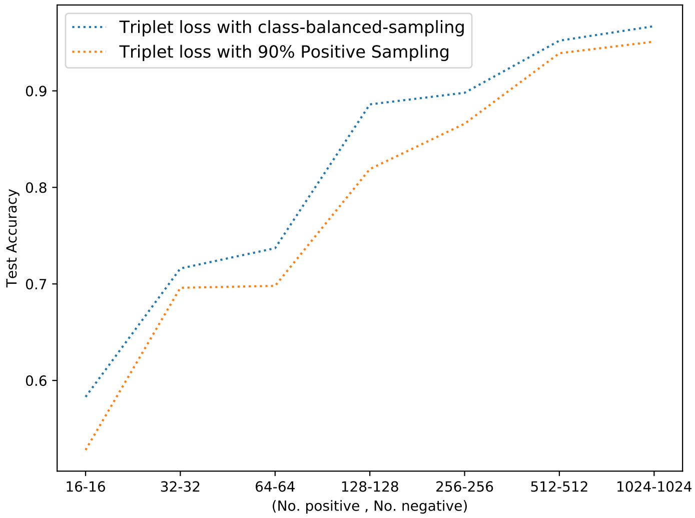


Figure A.1: *Comparison of performance of Triplet loss based classifiers in Balanced-Scratch Scenario.*

The impact of over-sampling. In Figure A.1 we observe that standard "Tripet loss" displays better performance than its over-sampling version in all experiments. In Figure A.2, we can see "Triplet loss + Positive Pair loss" also outperforms its over-sampling version in the smaller datasets, and they both give a similar performance in the larger datasets. We argue that over-sampling techniques are unnecessary when the dataset is already balanced.

The impact of different positive anchors Figure A.3 shows three variants of "Triplet loss + Positive Pair loss". "Triplet PA loss + Positive Pair PA loss" displays the worst test results in most of the experiments. "Triplet loss + Positive pair PA loss" has inferior results to "Triplet PA loss + Positive Pair loss" in smaller datasets; however,

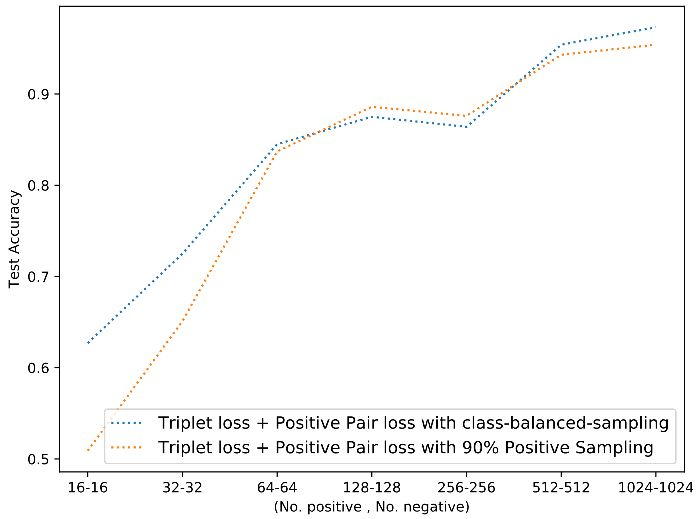


Figure A.2: *Comparison of performance of "Triplet loss + Positive Pair loss" based classifiers in Balanced-Scratch Scenario.*

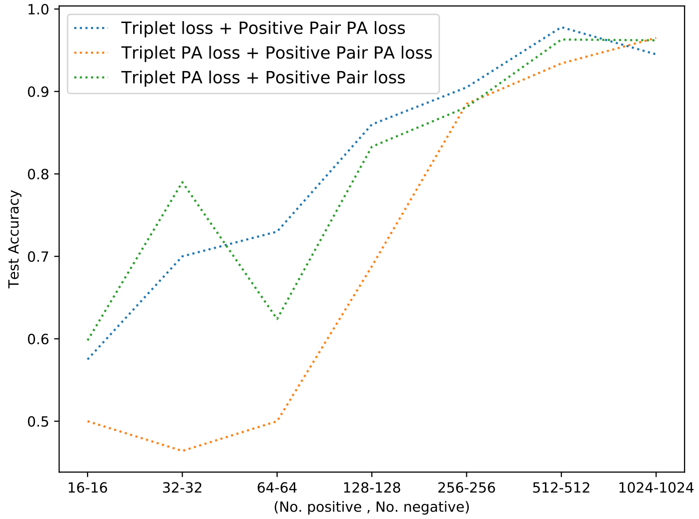


Figure A.3: *Comparison of performance of classifiers trained with "Triplet loss + Positive Pair loss" with different positive anchor considerations in Balanced-Scratch Scenario.*

they both demonstrate comparable performance in larger datasets. We argue that in the smaller balanced datasets, the class compactness is of greater importance than the margin between classes and, conversely, in the larger datasets, the maximized margin separating different classes dominates the classification results.

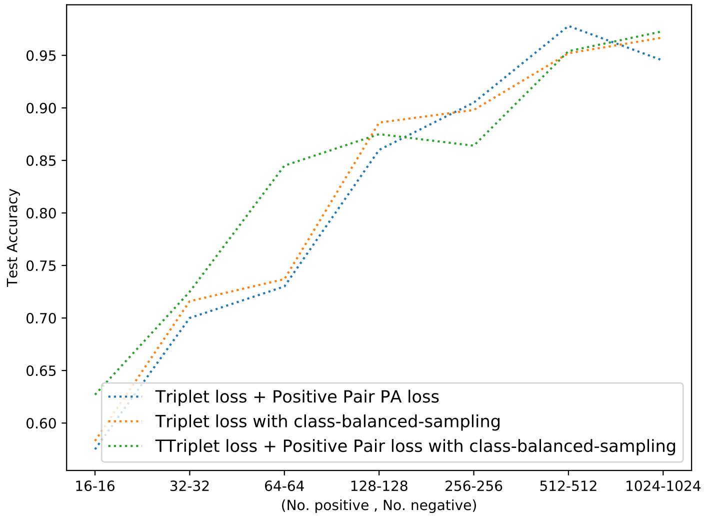


Figure A.4: *Comparison of performance of Triplet loss based classifiers in Balanced-Scratch Scenario.*

Summary of Triplet Methods. In Figure A.4, we compare the best models between "Triplet loss", "Triplet loss + Positive Pair loss" and "Triplet loss + Positive Pair PA loss". We notice that the performance of "Triplet loss + Positive Pair loss" tends to work better than the other two classifiers in smaller datasets, and they display close results in larger datasets. We thus choose "Triplet loss + Positive Pair loss" to represent the Triplet loss family when comparing with cross-entropy based methods.

Comparison of contrastive loss based classifiers. Figure A.5 depicts the test accuracy of contrastive losses. Most of them do not present a performance boost with additional data and remain under 70% accuracy in small-to-medium datasets. This pattern appears in all variants of contrastive loss, suggesting that these methods are not able to extract discriminate features from a small amount of data. When the dataset size increases, we find that contrastive loss based classifiers experience significant performance growth, except the "Positive Pair Anchor loss". We assume due to the exclusive focus on positive instances, "Positive Pair Anchor loss" loses the ability to recognize the negative examples, therefore leading to lower predictability for negative class. In summary, we choose the standard "contrastive loss" to represent the contrastive loss family.

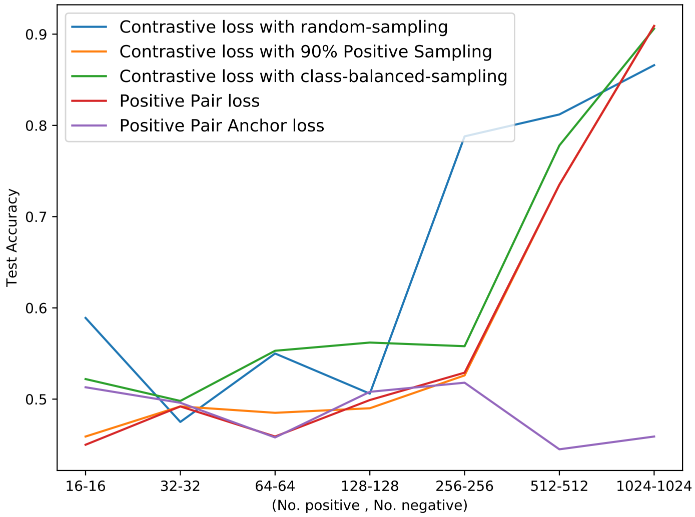


Figure A.5: *Comparison of performance of classifiers implementing Contrastive loss and its variants in Balanced-Scratch Scenario.*

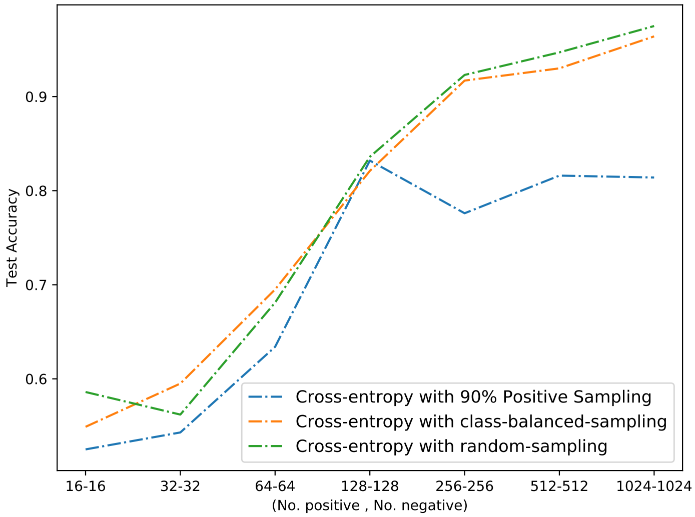


Figure A.6: *Comparison of performance of cross-entropy loss based classifier in Balanced-Scratch Scenario.*

Comparison of cross-entropy based classifiers. We can observe from Figure A.6 that three cross-entropy based classifiers have similar performance in the small datasets, but a noticeable difference can be viewed in larger datasets. “Cross-entropy with class-balanced-sampling” and “cross-entropy with random-sampling” have close performance in most experiments, meaning that sampling strategy has minimal impact on the balanced datasets. Additionally, both “cross-entropy with random-sampling” and “cross-entropy with class-balanced-sampling” outperform “cross-entropy with 90% positive sampling”. We argue that the use of over-sampling becomes trivial in larger balanced datasets because it disallows the classifier to extract useful features from the negative class. We choose “cross-entropy with class-balanced-sampling” to represent cross-entropy based classifiers.

A.2 Empirical Results with Balanced Data when Trained with Pretrained Parameters

We initialize our models with pretrained parameters from ImageNet to evaluate the impact of pre-training on balanced datasets. We evaluate different variants of each family and then compare across different methods to find out which method is more suitable for this scenario.

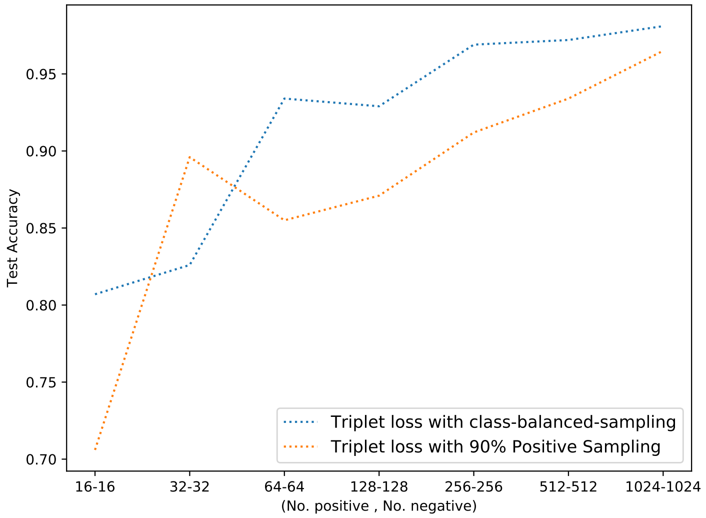


Figure A.7: *Comparison of performance of Triplet loss based classifiers in Balanced-Pretrained Scenario.*

The impact of over-sampling. From Figure A.7, we see that "Triplet loss" outperforms Triplet loss with over-sampling by a large margin. Similarly in Figure A.8, "Triplet loss + Positive Pair loss" maintains a clear margin with its over-sampling version in smaller datasets. We find that the over-sampling strategy impairs the model performance in balanced datasets because it excessively focuses on the positive data and consequently reduces the models' predictability of negative class.

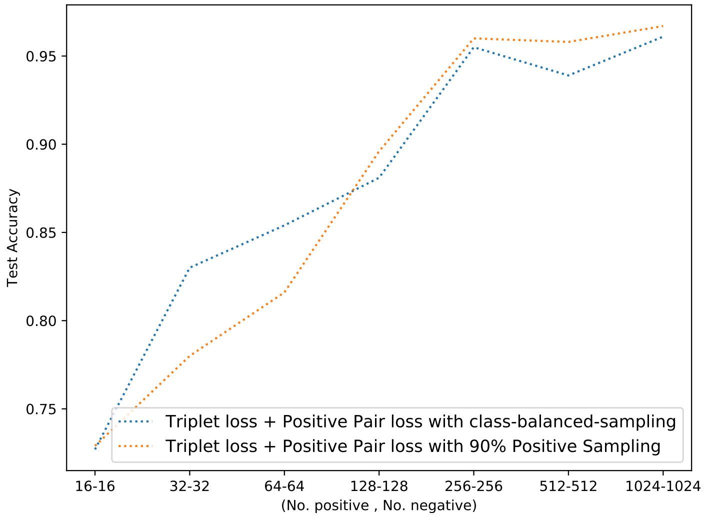


Figure A.8: *Comparison of performance of "Triplet loss + Positive Pair loss" based classifiers in Balanced-Pretrained Scenario*

The impact of different positive anchors. In Figure A.9, while the three classifiers demonstrate similar performance in larger datasets, their performances are very different in smaller datasets, especially the performance of "Triplet PA loss + Positive Pair PA loss". "Triplet PA loss + Positive Pair PA loss" performs the worst, i.e., with accuracy lower than 50%. It means that excessive attention on the positive examples reduces the model's learning on negative data. Such a classifier could retain its performance on larger datasets thanks to the unique design of the Triplet loss, which learns the distance information from positive-anchor triplets. In larger balanced datasets, though the excessive focus is given to the positive-anchor triplets, the model still can extract discriminative features from the negative-pair of the positive-anchor triplets because the larger dataset promotes the diversity of the negative instances. However, in smaller balanced datasets, the diversity of the negative examples is limited.

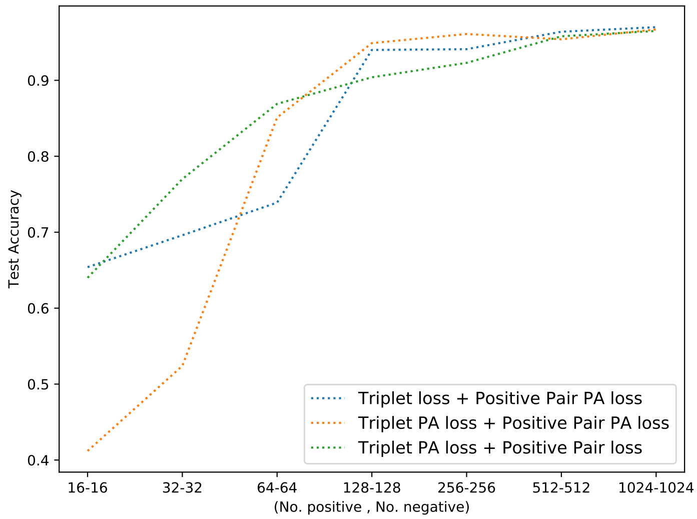


Figure A.9: *Comparison of performance of classifiers trained with "Triplet loss + Positive Pair loss" with different positive anchor considerations in Balanced-Pretrained Scenario.*

"Triplet PA loss + Positive Pair PA loss" is influenced by the information from the positive data only, mostly losing the ability to recognize the negative examples.

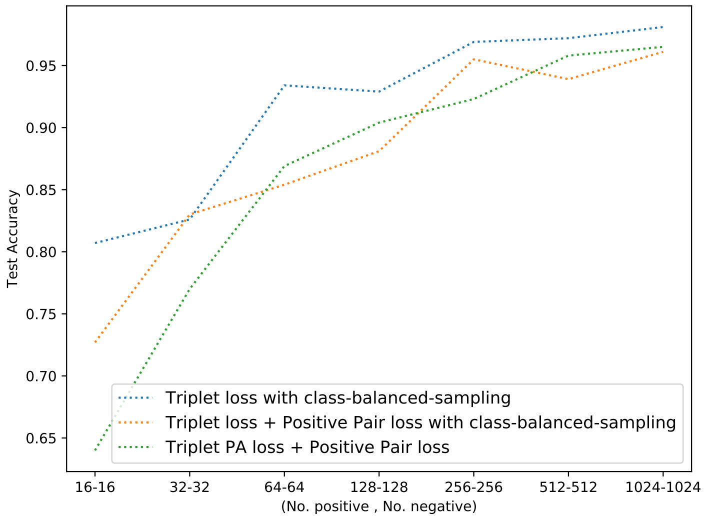


Figure A.10: *Comparison of performance of Triplet loss based classifiers in Balanced-Pretrained Scenario.*

Summary of Triplet Methods. From the previous analysis, we find the "Triplet loss", "Triplet loss + Positive Pair loss," and "Triplet PA loss + Positive Pair loss" are the best from each category. We thus plot them in Figure A.10, “Triplet loss” (in blue) has superior results than other methods. In the smallest dataset, it achieves more than 80% accuracy, outperforming the rest by a considerable margin. It sustains the outstanding performance in all experiments. In conclusion, the standard “Triplet loss” is the leading loss of the Triplet loss family in this scenario.

Comparison of contrastive losses based classifiers. We find in Figure A.11, “Positive Pair Anchor loss” is the best performing classifier, as it outperforms other methods considerably in small datasets. The second leading method is the “Contrastive loss”, depicted in blue, which yields a better result in the largest dataset with 1,024 examples per class. It is closely followed by “Positive Pair loss” (red curve). The remaining variants are less sensitive to the change of data size. Therefore, by comparing the performance of these methods, the “Positive Pair Anchor loss” is chosen as the representative of the contrastive loss family.

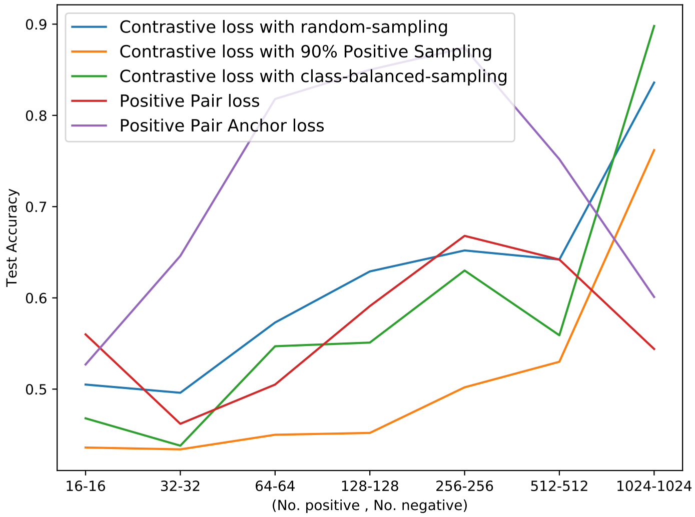


Figure A.11: *Comparison of performance of classifiers implementing contrastive loss and its variants in Balanced-Scratch Scenario.*

Comparison of cross-entropy based classifiers. We can see in Figure A.12 that “cross-entropy with class-balanced-sampling” and “cross-entropy with random-sampling” work much better than “cross-entropy with 90% positive sampling”. Considering that “cross-entropy with class-balanced-sampling” and “cross-entropy with random-sampling” demonstrate comparable performance, we choose the “cross-entropy with random-sampling" to represent cross-entropy loss based classifiers.

Comparison of the best classifier from each category. We plot the test accuracy of three leading classifiers chosen from each family in the above Figure A.13. It is clear that "Triplet loss" is the best performing method among the three classifiers as it displays superior results compared to the other two methods all the time. The second best method is the "cross-entropy with random-sampling", which has higher accuracy than the worst method by a substantial margin. We see that most Triplet loss based classifiers have close performance (see Figure A.10), and the worst classifier gives an accuracy of 65% in the smallest dataset, which is still higher than the performance of "cross-entropy with class-balanced-sampling" (see Figure A.13). As a result, we conclude Triplet losses based classifiers have better performance than cross-entropy based classifiers. Therefore, the metric-based method is the best choice in this scenario.

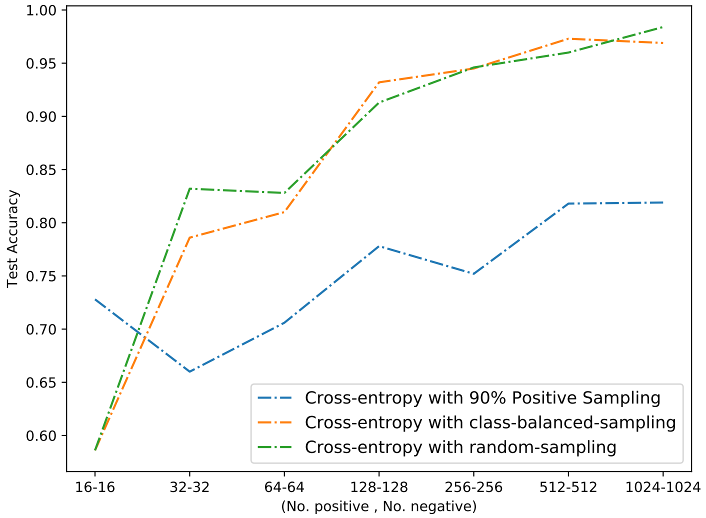


Figure A.12: *Comparison of performance of cross-entropy based classifiers in Balanced-Pretrained Scenario.*

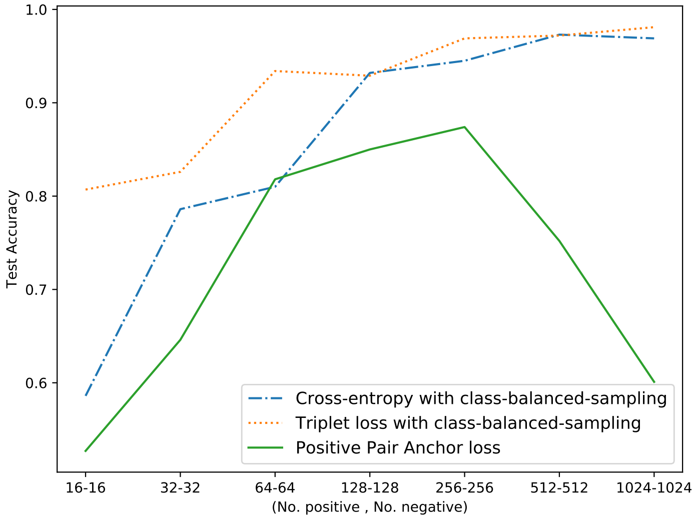


Figure A.13: *Comparison of performance of the best classifier from each category.*

A.3 Empirical Results with Imbalanced Data when Trained from Scratch

We conduct this set of experiments on imbalanced datasets with models trained from random initialization. We perform the evaluation firstly within different variants of each method and secondly across methods to analyze which method is the best.

The impact of over-sampling. We notice from Figure A.14 that “Triplet loss” tends to perform better than its over-sampling version, especially in the smallest dataset with 1,024 negative examples and 16 positive examples. In Figure A.15, we find that the over-sampling strategy does help with positive pair loss.

The impact of different positive anchors. In Figure A.16, the blue curve (“Triplet loss + Positive Pair PA loss”) has less appealing results in the smallest dataset, but it later displays comparatively stable growth. The other two classifiers are superior in smaller datasets, but experience strong oscillations in larger datasets. We conclude that ignoring all the negative-anchor triplets decreases the model’s robustness, particularly in larger imbalanced datasets. We use “Triplet loss + Positive Pair PA loss” to represent these three classifiers due to its stability.

Summary of Triplet Methods. We visualize the best Triplet losses based classifiers

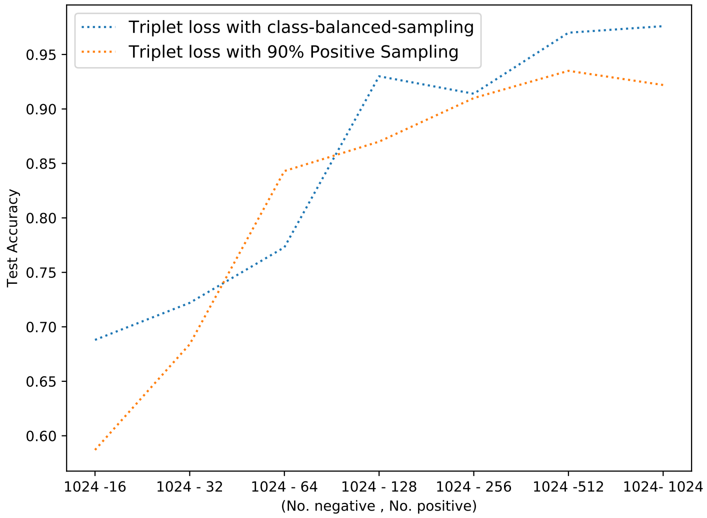


Figure A.14: *Comparison of performance of Triplet loss based classifier in Imbalanced-Scratch Scenario.*

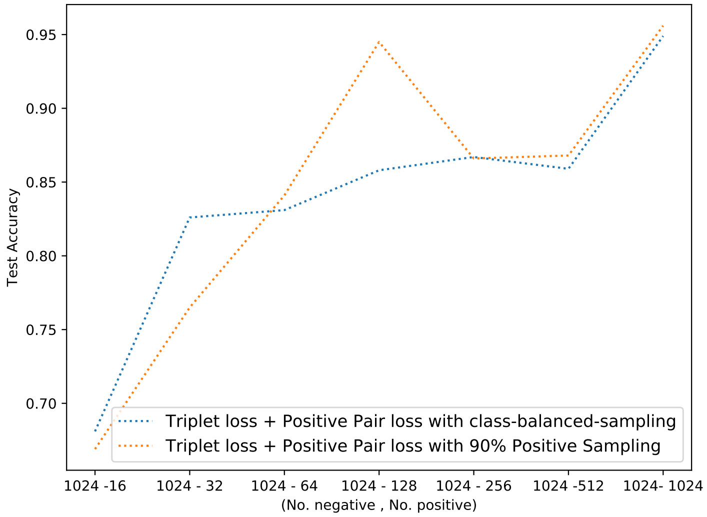


Figure A.15: *Comparison of performance of "Triplet loss + Positive Pair loss" based classifiers in Imbalanced-Scratch Scenario*

from preceding analyses in Figure A.17, the three classifiers exhibiting similar performances in small to medium datasets, but the "Triplet loss" showing better results in larger datasets. We argue that the oscillation of "Triplet loss + Positive Pair loss" with over-sampling results from substantially ignoring the negative examples in larger

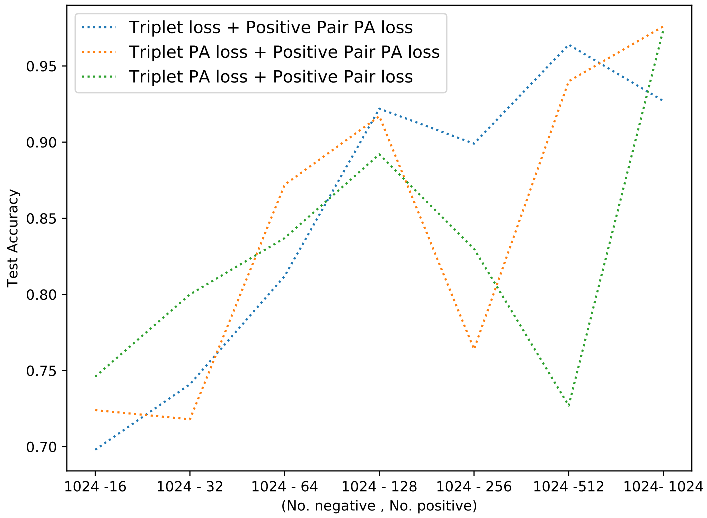


Figure A.16: *Comparison of performance of classifiers trained with "Triplet loss + Positive Pair loss" with different positive anchor considerations in Imbalanced-Scratch Scenario.*

datasets. We hence choose the "Triplet loss" to represent the Triplet loss family.

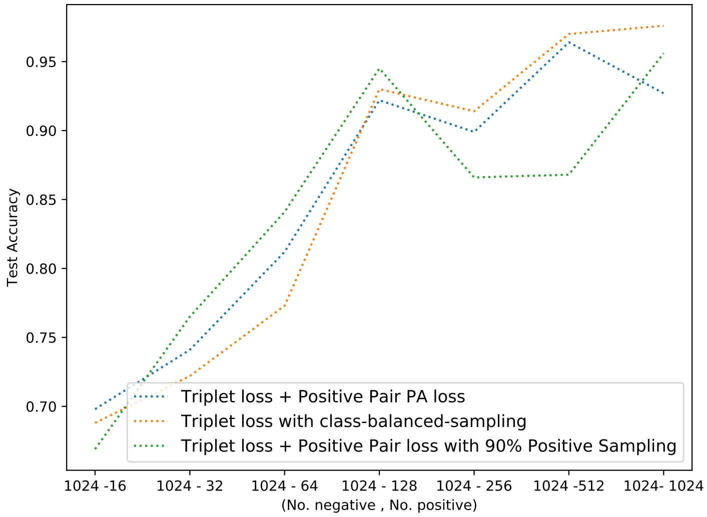


Figure A.17: *Comparison of performance of selected Triplet loss based classifiers in Imbalanced-Scratch Scenario.*

Comparison of contrastive losses based classifiers. In A.18, we see that all variants of contrastive losses in this scenario display ineffective performance increases with extra data in small datasets. We also observe that three of contrastive losses show sharp performance increases in larger datasets while the rest still display mild changes. We select the standard "Contrastive loss" to represent the contrastive loss family.

Comparison of cross-entropy based classifiers. We can observe from Figure A.19 that "cross-entropy with class-balanced-sampling" shows a better result than the other methods. Furthermore, a considerable difference is noticed between the curve of "cross-entropy with class-balanced-sampling" and the two other curves throughout experiments. Additionally, all three methods have significant increases with increasing positive samples. We hence conclude that "cross-entropy with class-balanced-sampling" works better in this scenario and we choose it to represent the cross-entropy based classifiers. Besides, we argue that balanced sampling is necessary given dataset imbalance.

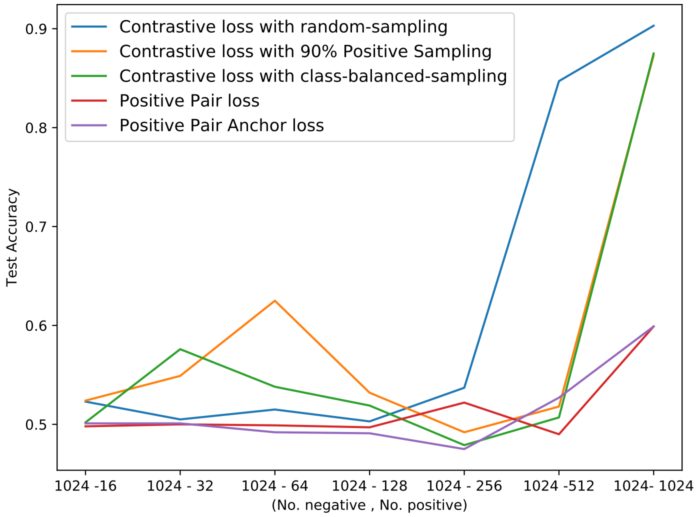


Figure A.18: *Comparison of performance of classifiers implementing contrastive loss and its variants in the Imbalanced-Scratch Scenario.*

Comparison of the best classifier from each category. It can be viewed in Figure A.20 that the "Triplet loss" classifier tends to work better than the cross-entropy and the contrastive loss based methods. Besides, we observe the "cross-entropy with class-balanced-sampling" classifier outperforms Triplet loss classifiers only in one experiment and remains inferior to it in the majority of experiments. "Contrastive loss with random-sampling" classifier displays the worst results in all experiments. Therefore, we conclude that the metric-based classifiers work better in this scenario where

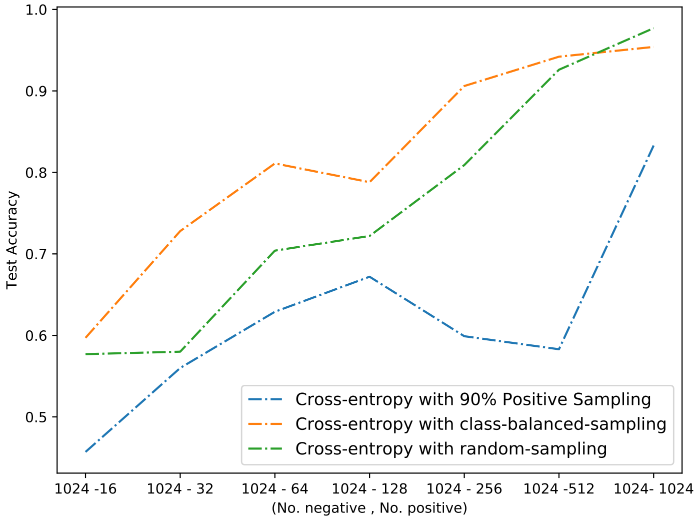


Figure A.19: *Comparison of performance of cross-entropy based classifiers in Imbalanced-Scratch Scenario.*

models are trained from scratch on imbalanced datasets.

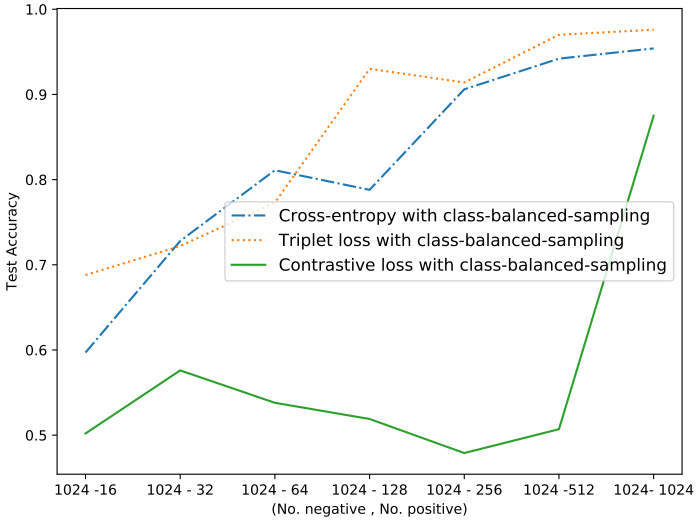


Figure A.20: *Comparison of performance of the best classifier from each category.*

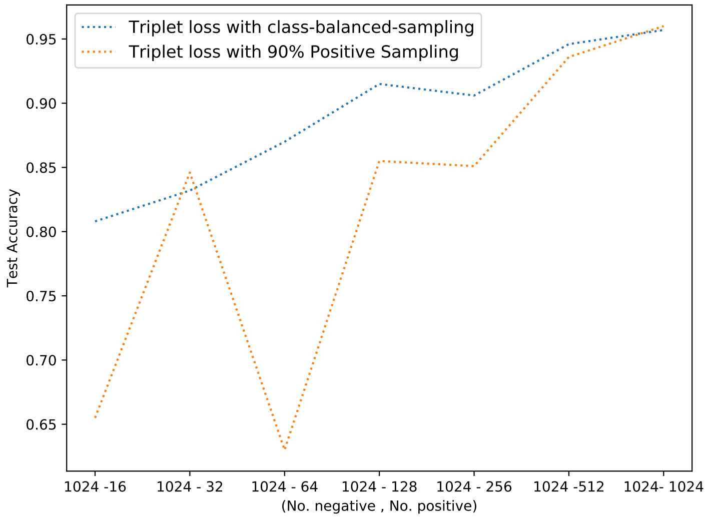


Figure A.21: *Comparison of performance of Triplet loss based classifier in Imbalanced-Pretrained Scenario.*

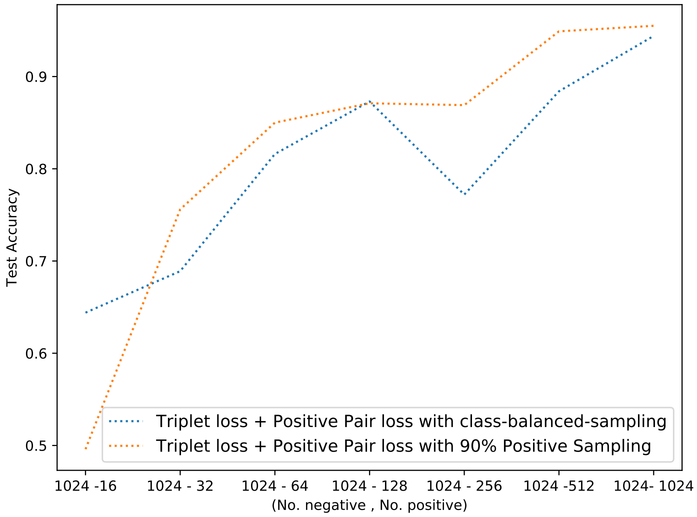


Figure A.22: *Comparison of performance of "Triplet loss + Positive Pair loss" based classifiers in Imbalanced-Pretrained Scenario*

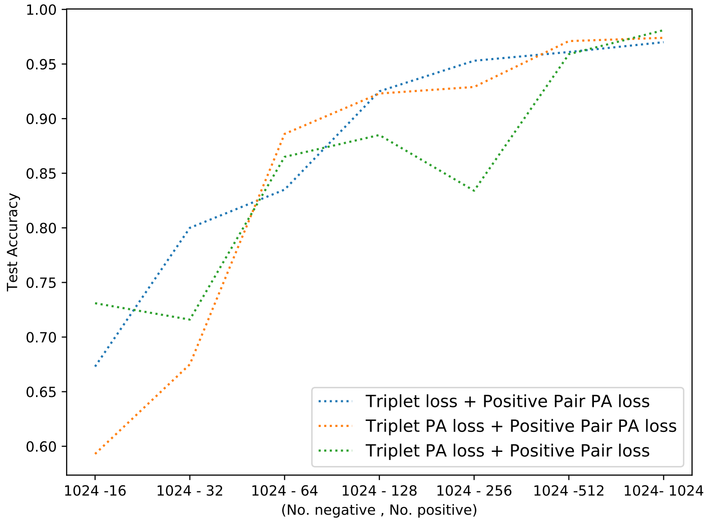


Figure A.23: *Comparison of performance of classifiers trained with "Triplet loss + Positive Pair loss" with different positive anchor considerations in Imbalanced-Pretrained Scenario.*

A.4 Empirical Results with Imbalanced Data when Trained with Pretrained Parameters

This set of experiments is performed with pretrained models on imbalanced datasets. The initial evaluation is done within each method to find the best-performing variant, and we then compare those best variants to conclude which method is the best in this scenario.

The impact of over-sampling. We can see from Figure A.21 that the standard "Triplet loss" performs the best. Its over-sampling version displays strong oscillations in small datasets. In Figure A.22 the "Triplet loss + Positive Pair loss" with over-sampling strategy outperforms the standard one in most experiments, and exhibits a strong and stable growth. The over-sampling strategy impacts classifiers differently, so it is difficult to conclude applicability to both cases.

The impact of different positive anchors. We also find that in Figure A.23 the three classifiers show comparable results in larger datasets. Additionally, "Triplet loss + Positive Pair PA loss" (in blue) is more stable than two other classifiers.

Summary of Triplet Methods. We again plot the best classifiers in Figure A.24.

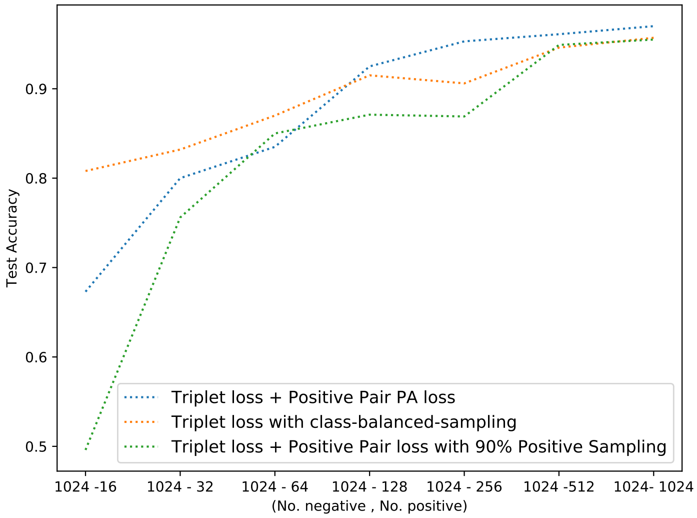


Figure A.24: *Comparison of performance of selected Triplet loss based classifiers in Imbalanced-Pretrained Scenario.*

We observe the “Triplet loss” (yellow curve) demonstrates excelling performance compared to the others in smaller datasets and maintains moderate competency overall. “Triplet loss + Positive Pair loss with 90% Positive Sampling” is the least performing one, with down to 50% accuracy in the smallest dataset. However, it exhibits strong growth and displays comparable results in larger datasets. We conclude that the standard Triplet loss works best because of its stability.

Comparison within contrastive loss based classifiers. It can be observed in Figure A.25 that “Contrastive loss with random-sampling” and “Contrastive loss with class-balanced-sampling” are more stable than other variants. Their performances are close, and they improve as the dataset size increases. Meanwhile, “Contrastive loss with 90% positive sampling” and two variants of “Positive Pair loss” exhibit oscillations in all experiments, suggesting the models’ instabilities in this scenario. Since “Contrastive loss with random-sampling” and “Contrastive loss with class-balanced-sampling” have comparable performance, we choose “Contrastive loss with random-sampling” as the representative of the contrastive loss family.

Comparison of cross-entropy based classifiers. In Figure A.26, we can observe “cross-entropy with class-balanced-sampling” outperforms the “cross-entropy with

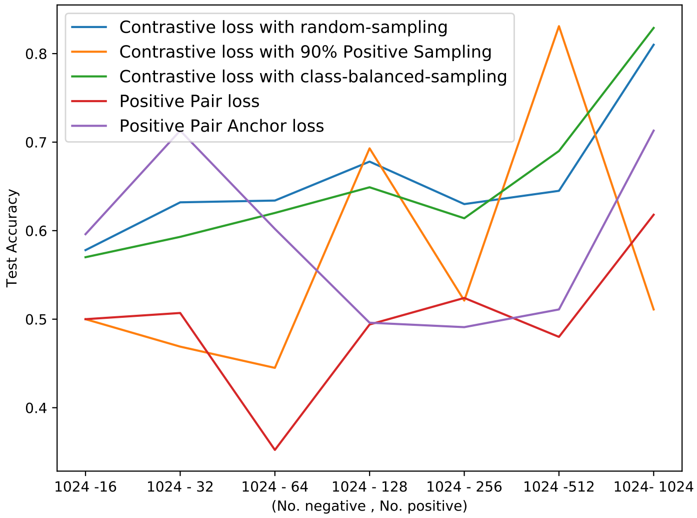


Figure A.25: *Comparison of performance of classifiers implementing contrastive loss and its variants the Imbalanced-Pretrained Scenario.*

random-sampling" by a large margin. Furthermore, the "cross-entropy with random-sampling" performs better than the "cross-entropy with 90% positive sampling". "Cross-entropy with class-balanced-sampling" and "cross-entropy with random-sampling" display a positive increase in most cases, and, in the meantime, "cross-entropy with random-sampling" has witnessed a more rapid increase. In contrast, ""cross-entropy with 90% positive sampling suffers several oscillations. Therefore, we conclude that "cross-entropy with class-balanced-sampling" is the most robust method in this situation.

Comparison of the best classifier from each category. We choose the best classifier from each family and visualize classifiers' performance in Figure A.27. It is clear that "cross-entropy with class-balanced-sampling" is the leading classifier, outperforming other classifiers in the group. Both "cross-entropy with class-balanced-sampling" and "Triplet loss" outperform the "Contrastive loss" remarkably. Thus, we claim that "cross-entropy with class-balanced-sampling" with the pretrained parameters can handle the imbalance more effectively than metric-based classifiers.

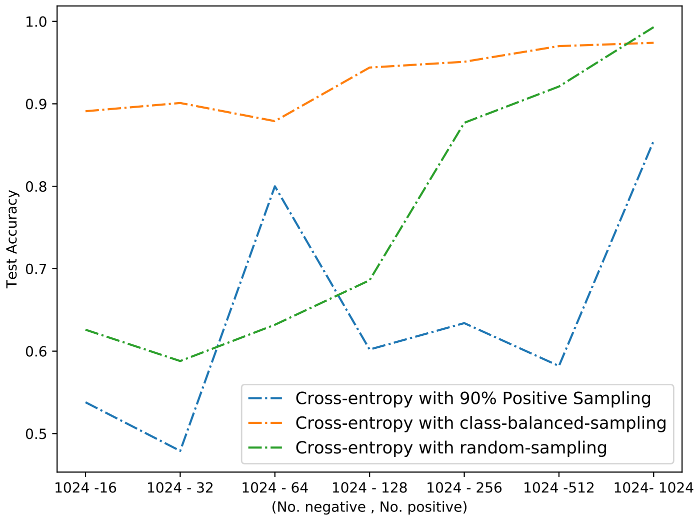


Figure A.26: *Comparison of performance of cross-entropy based classifiers in the Imbalanced-Pretrained Scenario.*

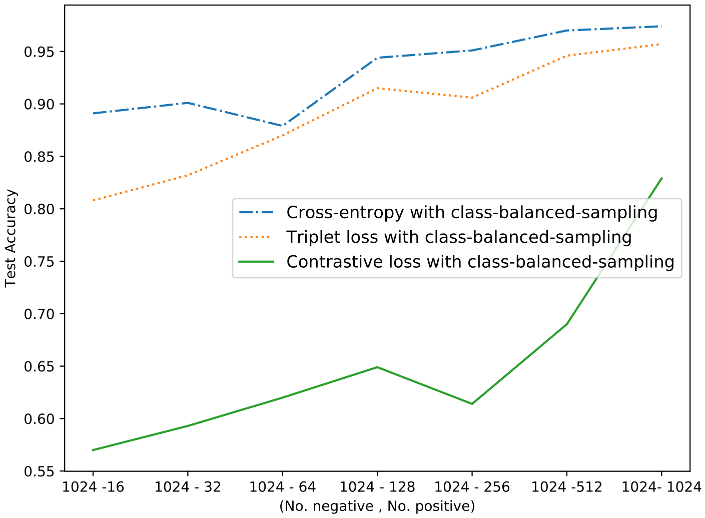


Figure A.27: *Comparison of performance of the best classifier from each category.*