AUDIO CLASSIFICATION WITH FEW-SHOT LEARNING

BİRKAÇ ÖRNEKLİ ÖĞRENME İLE SES SINIFLANDIRMA

ENES FURKAN ÇİĞDEM

ASSOC. PROF. DR. HACER YALIM KELEŞ

Supervisor

Submitted to

Graduate School of Science and Engineering of Hacettepe University

as a Partial Fulfillment to the Requirements

for the Award of the Degree of Master of Science

in Computer Engineering

August 2024

ABSTRACT

AUDIO CLASSIFICATION WITH FEW-SHOT LEARNING

Enes Furkan ÇİĞDEM

Master of Science, Computer Engineering

Supervisor: Assoc. Prof. Dr. Hacer YALIM KELEŞ

August 2024, 67 pages

This thesis does a full experimental study of the few-shot classification problem in the audio domain to compare how well episodic and non-episodic training methods work. Three different optimization algorithms are trained with the non-episodic method, and the effect of the training techniques on the classification performance is investigated. In making these comparisons, simple feature transformations have been employed to improve performance, and their effect on performance has been analyzed.

The few-shot audio classification task has been conducted in scenarios with limited data. This study uses two distinct data sets: Environmental Sound Classification - 50 and Google Speech Commands. ESC-50 includes environmental non-speech noises. GSC encompasses basic spoken orders. Three distinct scenarios are constructed in which the amount of training data is constrained for each data set by selecting 5, 10, and 15 samples per class. A series of comprehensive experiments have been conducted with these different training sets using three different optimization models in non-episodic experiments: single-stage hybrid loss optimization (SSHLO), single-stage loss optimization (SSLO), and two-stage loss optimization (TSLO). The results of these experiments are then compared between the three optimizations and episodic training.

The findings of our research point out that the non-episodic training approach is more effective than the episodic training approach in the audio domain when used with a pre-trained model. In terms of optimizations, the results demonstrate that single-stage hybrid loss optimization (SSHLO) is the most superior optimization on the two data sets.

Keywords: Audio Processing, Audio classification, Episodic Training, Non-episodic Training, Few-shot Learning, Few-shot Classification, Contrastive Learning, Simple Feature Transformations, Neural Speech Embedding Model

ÖZET

BİRKAÇ ÖRNEKLİ ÖĞRENME İLE SES SINIFLANDIRMA

Enes Furkan ÇİĞDEM

Yüksek Lisans, Bilgisayar Mühendisliği

Danışman: Assoc. Prof. Dr. Hacer YALIM KELEŞ

Ağustos 2024, 67 sayfa

Bu tez, epizodik ve epizodik olmayan eğitim yöntemlerinin ne kadar iyi çalıştığını karşılaştırmak için ses alanındaki birkaç vuruşlu sınıflandırma probleminin tam bir deneysel çalışmasını yapmaktadır. Üç farklı optimizasyon algoritması epizodik olmayan yöntemle eğitilmiş ve eğitim tekniklerinin sınıflandırma performansı üzerindeki etkisi araştırılmıştır. Bu karşılaştırmalar yapılırken, performansı artırmak için basit özellik dönüşümleri kullanılmış ve bunların performans üzerindeki etkisi analiz edilmiştir.

Az sayıda ses sınıflandırma görevi, sınırlı veriye sahip senaryolarda gerçekleştirilmiştir. Bu çalışmada iki farklı veri seti kullanılmıştır: Çevresel Ses Sınıflandırması - 50 ve Google Konuşma Komutları. ESC-50 çevresel konuşma dışı sesleri içerir. GSC temel sözlü emirleri kapsar. Eğitim verisi miktarının her veri seti için sınıf başına 5, 10 ve 15 örnek seçilerek kısıtlandığı üç farklı senaryo oluşturulmuştur. Epizodik olmayan deneylerde üç farklı optimizasyon modeli kullanılarak bu farklı eğitim setleriyle bir dizi kapsamlı deney gerçekleştirilmiştir: tek aşamalı hibrit kayıp optimizasyonu (SSHLO), tek aşamalı kayıp optimizasyonu (SSLO) ve iki aşamalı kayıp optimizasyonu (TSLO). Bu deneylerin sonuçları daha sonra üç optimizasyon ile epizodik eğitim arasında karşılaştırılmıştır.

Araştırmamızın bulguları, önceden eğitilmiş bir modelle birlikte kullanıldığında ses alanında epizodik olmayan eğitim yaklaşımının epizodik eğitim yaklaşımından daha etkili olduğuna işaret etmektedir. Optimizasyonlar açısından, sonuçlar tek aşamalı hibrit kayıp optimizasyonunun (SSHLO) iki veri seti üzerinde en üstün optimizasyon olduğunu göstermektedir.

Keywords: Ses İşleme, Ses sınıflandırma, Bölümsel eğitim, Bölümsel olmayan eğitim, Epizodik eğitim, Epizodik olmayan eğitim, Birkaç örnekli öğrenme, Birkaç örnekli sınıflandırma, Karşılaştırmalı öğrenme, Basit özellik dönüşümleri, Sinirsel ses gömüleme modeli

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Assoc. Dr. Hacer YALIM KELEŞ for her guidance and support during this research.

Moreover, I thank my thesis defense jury members Prof. Dr. Asım Egemen YILMAZ and Asst. Prof. Dr. Engin DEMİR for their time, effort, and insightful feedback.

I am deeply grateful to my loving wife, İrem, for her unwavering support, understanding, and constant encouragement. Her love and support made this journey possible.

I would like to thank Sestek, especially my team leader Mustafa Erden, for their support and flexibility during my master's degree journey.

CONTENTS

Page

TABLES

FIGURES

ABBREVIATIONS

SSLO	:	**S**ingle **S**tage **L**oss **O**ptimization
SSHLO	:	**S**ingle **S**tage **H**ybrid **L**oss **O**ptimization
TSLO	:	**Two S**tage **L**oss **O**ptimization
ESC-50	:	**E**nvironmental **S**ound **C**lassification - 50
GSC	:	**G**oogle **S**peech **C**ommands
ECAPA	:	**E**mphasized **C**hannel **A**ttention, **P**ropagation and **A**ggregation
MLP	:	**M**ulti **L**ayer **P**erceptron
SSCL	:	**S**upport **S**upport **C**ontrastive **L**earning
PQCL	:	**P**rototype **Q**uery **C**ontrastive **L**earning
CE	:	**C**ross **E**ntropy
SPC	:	**S**ample **P**er **C**lass

1. INTRODUCTION

In the rapidly evolving field of machine learning, conventional and fully supervised learning methodologies have demonstrated efficacy in a range of domains. However, these approaches typically necessitate the availability of extensive labeled data sets. It is not always feasible to obtain a sufficiently large data set to address real-world problems using fully supervised learning methods. In many cases, data is scarce, which presents a significant challenge. For example, it could be challenging to collect data for a local language adaptation used for spoken language understanding tasks in a call center. Similarly, the use of user-defined custom keywords for keyword spotting may raise privacy concerns. Therefore, dealing with limited data is important in the auditory domain.

1.1. Overview

A few-shot classification aims to adopt a few samples for each class in a data set. Audio classification with few-shot learning is considered a subtask for few-shot classification working on auditory events.

This thesis investigates several aspects of few-shot audio classification. In particular, it utilizes the Environmental Sound Classification ESC-50 and Google Speech Commands (GSC) data sets. In addition, it compares various training approaches to determine the optimal approach for limited training data. Furthermore, this comparison, contrasts different loss optimization types across different data scarcity levels.

1.2. Motivation

Few-shot learning aims to adapt new tasks to previous experiences as humans do. Many problems have successfully employed few-shot learning. On the other hand, traditional fully supervised approaches need large labeled datasets to produce preferable results. In most real-life situations, a large data set is not available. Additionally, the real-life data sets may

be so limited that only a few examples of classes are available in the data set, which has a restricted number of samples in the training stage.

In the audio domain, it may not always be possible or logical to have large amounts of labeled data. Many reasons lead to data scarcity problems, such as privacy concerns or labeling costs.

At this moment, with limited data, it is crucial to determine how the model should be trained by including two different types of losses. The first objective of this thesis is to compare different loss optimization procedures to be able to settle optimal loss functions to be used during training to obtain a successful model.

1.3. Contributions

- A novel study has been conducted to compare different optimization approaches utilizing non-episodic training via comprehensive experiments in the audio domain. This study focused on both speech and non-speech environmental sound data sets and aimed to modify a pre-trained model using a very large data set. Furthermore, the impact of basic feature modifications on classification accuracy has been examined.

- It has been proposed that the concurrent application of unsupervised contrastive loss and supervised loss optimization improves few-shot classification on both speech and non-speech datasets. This approach can be employed in both non-episodic [3] and episodic training frameworks. Furthermore, comparisons have been conducted with alternative forms of loss enhancement. Additionally, the impact of simple transformations to features on the classification performance is examined.

- This thesis presents important observations between optimizations using non-episodic training and ProtoNets using episodic training. The findings demonstrate that, in the presence of a large-scale pre-trained model, non-episodic training yields more effective results, even though it employs a less complex structure.

1.4. Organization

The organization of the thesis is as follows:

- **Chapter 1** states the motivation, contributions, and the scope of the thesis.
- **Chapter 2** provides background information about methodologies utilized for audio classification with few-shot learning framework in this thesis.
- **Chapter 3** describes the related works in the literature.
- **Chapter 4** introduces methodology.
- **Chapter 5** presents experimental findings and discussion.
- **Chapter 6** provides a concise overview of the thesis.

2. BACKGROUND OVERVIEW

2.1. Classification

Classification is one of the fundamental methodologies in the machine learning field. This task involves categorizing data samples into predefined classes according to the extracted attributes of the samples. As a supervised machine learning methodology, it has been employed in various applications such as image recognition [4] and audio classification [5].

2.2. Audio Classification

Audio modality encompasses all forms of sound data, including speech and environmental sounds perceived by humans or electronic devices. Audio classification is a subtask of the classification task family that involves mapping audio signals into predefined categories. Through the process of audio signal categorization, it becomes feasible to gain a deeper comprehension of the fundamental signal, its organization, and its substance, thereby facilitating a wide range of practical uses with many applications on various problems such as environmental sound classification [6], spoken intent classification [7], and speaker recognition [8].

2.3. Sound Representations

An audio signal is a type of signal that carries information within the frequency range of 20 Hz to 20 kHz and is perceptible to the human ear. The process of audio representation entails the extraction of attributes or features from an audio source to accurately represent its composition.

2.3.1. Raw Waveform

A waveform represents the variation in amplitude of a signal over time. When an audio signal is recorded, it must be transformed from a sound wave into an electrical signal. The voltage of this electrical signal fluctuates. These differences are directly related to the fluctuations in air pressure caused by the initial sound wave.

An analog-to-digital converter (ADC) performs periodic sampling on the electrical signal. It records individual points that indicate the intensity of the signal at particular instances in time. The given samples are graphed, with the horizontal axis representing time and the vertical axis representing amplitude.

2.3.2. Mel Filterbanks

A spectrogram is a fundamental audio representation that depicts frequency changes over time by applying a short-time discrete Fourier transform (STDFT) to an audio utterance. The Mel spectrogram is a common type of spectrogram that is scaled by the Mel scale. Mel scaling is proposed to scale audio signal frequency to make frequency changes more perceivable by converting frequency into units which mean equidistant units in frequency are equidistant in tone for humans.

Mel-filterbanks are generated using a predefined number of filter bands. These bands are very useful for retrieving perceptually meaningful splits in audio. Since its success in getting meaningful splits and being closely aligned with the human auditory system, it is frequently used in audio and speech tasks.

When it comes to transformations, although audio modality is similar to image modality, image transformations like cropping and rotation do not work for audio. Thus, audio representations need to be transformed by specific transformations like pitch shifting or time shifting.

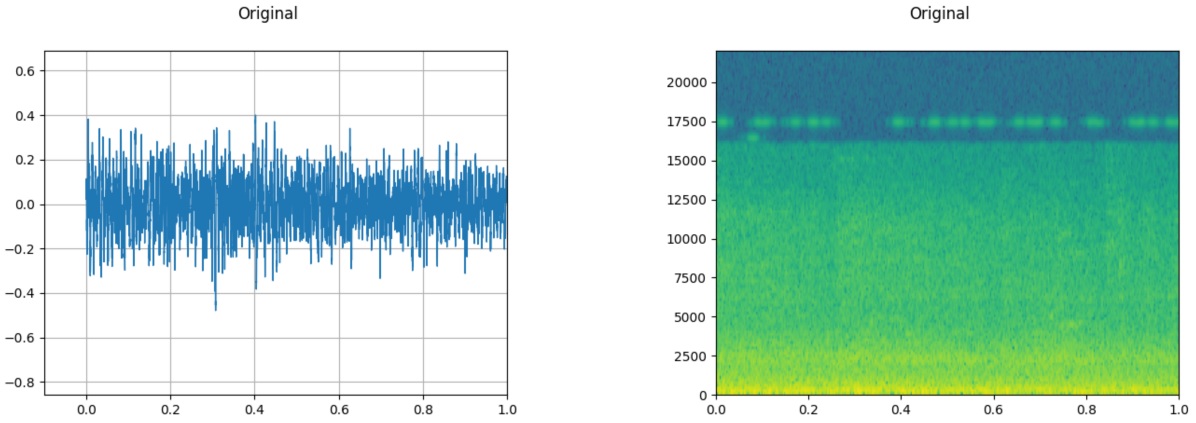

Figure 2.1 Images show the representation of a train class sample from ESC-50. (a) and (b) images present raw waveform and mel-spectrogram, respectively.

2.4. Few-Shot Learning

Few-shot learning is an evolving framework that aims to train successful AI models dealing with a limited number of samples. In machine learning, few-shot learning plays a crucial role in coping with insufficient data for training.

2.4.1. Episodic Training

Conventionally, many few-shot learning approaches [9–11] utilize episodic training regime. In this training regime, training data is organized into episodes that replicate few-shot scenarios, allowing the model to learn effectively from limited examples and generalize to novel classes. In a typical few-shot classification task, episodes are constructed by randomly sampling a small number of classes and examples to form support sets and query sets. In the support set, there are a limited number of examples for the chosen classes. These samples are utilized to train models to learn the characteristics of classes. The query set also includes a limited number of examples from the same classes in the support set that are employed to measure and enhance model fit.

This type of training is structured into a series of episodes that are designed to mimic test conditions. Each episode includes two different sets: the support set and the query set.

The **support set** is a small set that is used for model adaptation. Typically, it consists of K samples for N classes in an N-way K-shot setting. For instance, 5 samples exist for 5 classes in a 5-way 5-shot setting. The **query set** is a separate set that consists of Q samples for the same N classes in the support set. The model first learns from the support set. Subsequently, the query set is used to evaluate the model performance.
More formally, each episode is constructed as follows:

$$S = \{(\mathbf{x}_1^1, y_1^1), (\mathbf{x}_1^2, y_1^2), \ldots, (\mathbf{x}_N^K, y_N^K)\}$$

$$Q = \{(\mathbf{x}_1^1, y_1^1), (\mathbf{x}_1^2, y_1^2), \ldots, (\mathbf{x}_N^q, y_N^q)\}$$

$$Episodes = \{(S_1, Q_1), \ldots, (S_E, Q_E)\}$$

where N and K are the number of classes in support and the number of samples per class in the support set. The number of samples in the query set is represented as q.

2.4.1.1. Prototypical Networks

Prototypical Networks (ProtoNets) [9] are a well-known few-shot learning approach that is designed to cope with classifying unseen categories with limited labeled examples. The core idea of ProtoNets is learning a metric space where the distance between points from the same class is minimized, and the distance between points from different classes is maximized. ProtoNets uses an embedding network to capture important features from the input data samples. A prototype is simply a mean of representation vectors calculated from support set samples using the embedding function. Once feature embeddings are extracted for samples for each class in the support set. The mean representation vector is obtained as given in (Eqn. 1) where g denotes the embedding network, S_c denotes the support set, and P_c denotes the prototype vector.

$$\mathbf{P}_c = \frac{1}{|S_c|} \sum_{(x_k, y_k) \in S_c} g(x_k) \tag{1}$$

Subsequently, the distances between query set samples and class prototypes are calculated using typically squared Euclidean distance. The closest prototype class is assigned as a predicted class for a query sample as given in Eqn. (2).

$$P(y = c|\hat{x}) = \text{softmax}\left(-d\left(g(\hat{x}), \mathbf{P}_c\right)\right) = \frac{e^{-d(g(\hat{x}),\mathbf{P}_c)}}{\sum_{\hat{c}\in\mathcal{C}} e^{-d(g(\hat{x}),\mathbf{P}_{\hat{c}})}} \tag{2}$$

ProtoNets utilize a negative log-likelihood loss function to optimize embedding network. The aim is to minimize the probability of incorrectly classifying a query instance. By minimizing this loss, the model learns an embedding space where query points are close to the prototypes of their correct class and far from those of incorrect classes.

$$\mathcal{L}_{\text{ProtoNet}} = -\log P(y = c \mid \hat{x}) \tag{3}$$

2.4.2. Non-episodic Training

Non-episodic training for few-shot classification involves training methods that differ from the common traditional episodic training approach used in the few-shot learning area. In this study, the training schema utilized by Wang et al. [12] is employed for all non-episodic experiment settings. This training regime employs a pre-trained deep network on base classes to obtain feature embeddings. It then applies centering and L2 normalization to the resulting features of novel classes for evaluation with the nearest neighbor classifier with Euclidean distance. This straightforward method has demonstrated significant enhancements when compared to meta-learning alternatives in the image domain.

Non-episodic training involves methods that do not utilize support and query sets, unlike conventional episodic training. The model learns from a continuous stream of data batches, like in conventional supervised learning, and it does not utilize episodic sampling. A data batch consists of input and output pairs.

$$\mathcal{D} = \{(\mathbf{x}_0, y_0), (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$$

$\mathcal{D}$ denotes the dataset consisting of N training examples, where $\mathbf{x}_i$ represents the feature vector of the i-th sample, and y_i is the corresponding label.

2.5. Contrastive Learning

Contrastive learning is a technique used in self-supervised machine learning, especially in unsupervised environments where labeled data is scarce. A core idea of this approach is teaching a model to discriminate between similar and dissimilar data points. It uses the method of creating groups of data that can be either similar or dissimilar. For instance, considering audio classification, an audio utterance could be augmented in several ways, such as time shifting or noise addition, while a completely different audio utterance would be the dissimilar pair. It learns from relative differences and similarities between data rather than relying solely on explicit labels. This helps the models develop a more nuanced understanding of the data.

2.5.1. SimCLR: Simple Framework for Contrastive Learning of Visual Representations

SimCLR [2] is a self-supervised learning framework that simplifies contrastive learning. The framework comprises four principal components. Firstly, data augmentation is utilized to generate two correlated views of the same image through the application of random transformations. Secondly, a base encoder is employed as a fundamental component that is expected to learn the representation of the system. Originally, ResNet50 [4] was employed for the extraction of representation vectors. Then, a projection head that is an MLP is utilized to map the high-dimensional representation vectors from the base encoder to a lower-dimensional space to apply contrastive loss. The contrastive loss function (NT-XENT) plays a crucial role in maximizing the agreement between similar pairs of images (positive pairs) and separating them from other images in the batch (negative pairs).

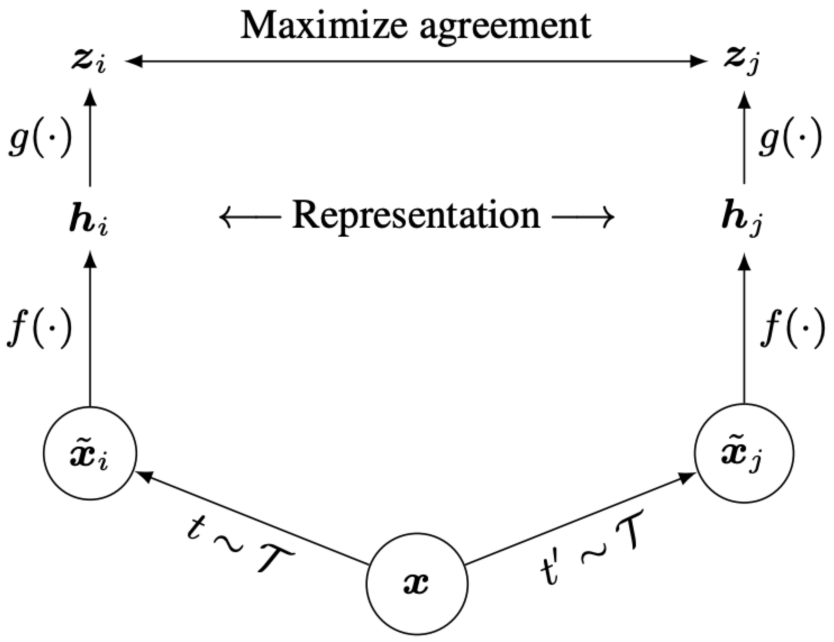

Figure 2.2 SimCLR standard pipeline. Image source:[2]

2.6. Neural Embedding Models

Neural embedding models are powerful techniques to conduct the transformation of high-dimensional data, such as textual or audio, into their representations in a lower-dimensional space [13–16]. They capture semantic relationships in the high-dimensional data and represent them in a compact vector form to be able to perform various downstream tasks such as classification, clustering, and retrieval. Regarding its applications in the field of audio, neural speech embeddings have significantly impacted speaker verification [17], speaker diarization [18], text-to-speech [19] and speech classification [20–22] tasks.

3. RELATED WORK

3.1. Self-supervised Learning

There are many studies [23, 24] on this subject that aim to learn from the data itself by using unlabeled data, which is especially valuable in cases of data scarcity. SimSiam [25] introduces a significant study on the capabilities of Siamese networks in unsupervised visual representation learning positive pairs without the necessity for negative sample pairs, large batches, or momentum encoders. It presents a simple approach to Siamese networks to learn directly to maximize the similarity between two augmentations of a single image. SimCLR [2] by Chen et al. eliminates the need for specialized architectures or memory banks. It achieves high performance by employing a simple methodology utilizing two main components: a series of data augmentations to generate variations of the same image and a contrastive loss function to maximize agreement between these variations. The Barlow Twins [26] study aims to ensure independence between the features by maximizing the diagonal of the correlation matrix.

The Audio Barlow Twins [27] is a self-supervised audio representation learning method that proposes an adaptation of the Barlow Twins to the audio domain. This novel method aims to overcome the limitations of current self-supervised learning techniques that require negative samples or asymmetric learning updates. By utilizing a cross-correlation matrix to force the embeddings of augmented views of audio data towards the identity matrix, this method ensures that similar instances are embedded near each other while minimizing redundancy in the embedding components. The method is pre-trained on the large-scale AudioSet data set and evaluated on many downstream tasks, such as speech and environmental sound classification. It provides 78.6% accuracy on the ESC-50 data set. COLA [28] is a self-supervised, contrastive learning-based framework by Saeed et al. for developing general-purpose audio representations. COLA exploits similarities between audio segments from the same recording and differences with segments from other recordings. COLA utilizes pre-training on a large-scale data set and uses learned representations for diverse audio

classification tasks such as speech, music, and animal sounds, providing high-accuracy results.

Haider Al-Tahan et al. [29] investigate that the utilization of contrastive learning to auditory data can enhance auditory representations. They combine supervised and contrastive learning to improve efficiency and speed of training and achieve better performance with less number of labeled data compared to conventional supervised methods. SoundCLR [30] have achieved important results using the SimCLR structure. In this study, the authors applied hybrid supervised and comparative loss training using the ESC-50 data set. It is aimed at learning effective representations only from environmental sound data without resorting to few-shot methods.

3.2. Few-Shot Classification

Few-shot learning is another important topic, like self-supervised learning when labeled data is limited. In particular, non-parametric metric-based methods using episodic learning are approaches worth mentioning. Jake Snell et al. [9] introduce prototypical networks, a simple strategy to tackle few-shot image classification learning in a metric space where the classification is carried out using the nearest centroid classifier mechanism. Oriol Vinyals et al. [10] address the challenge of one-shot learning in few-shot learning, where a model learns one sample per class. Furthermore, they investigate one-shot learning for language modeling.

Lim et al. [31] approaches to few-shot image classification incorporating contrastive learning. The authors propose a method that addresses the challenge of generalizing from a limited amount of samples by enriching model representations with multiple self-supervision objective functions. They compare and discuss the effects of cross-entropy loss and contrastive loss combinations. They demonstrate significant results on benchmark data sets in the image domain. They highlight the effectiveness of combining multiple self-supervision losses and complex augmentations to strengthen the generalization capabilities of few-shot learning models by considering the accuracy values obtained. [32] proposes an approach integrating self-supervised learning, prototypical networks, and knowledge distillation

to enhance few-shot classification success on benchmark image classification data sets, including miniImageNet, tieredImageNet, and CIFAR-FS. There are three stages applied in this method: pre-training, fine-tuning, and self-distillation. The pre-training phase employs self-supervised learning to enhance sample discrimination. This initial training targets the model's generalized weights. The fine-tuning stage integrates both self-supervised and few-shot losses, and it tries to prevent over-fitting and maintain embedding diversity. During this fine-tuning phase, the model minimizes the distance between support and query samples and adapts to few-shot tasks. Finally, a teacher-student architecture is employed in the self-distillation stage. The model, trained in the fine-tuning stage, is utilized as a teacher model to enable student model performance improvement by reducing overfitting.

Wang et al. [12] explore the accuracy of nearest-neighbor baselines utilizing non-episodic training without meta-learning. Moreover, they demonstrate that simple feature transformations can achieve competitive few-shot learning accuracy using a pre-trained deep network, outperforming prior results in specific settings in the image domain. Tian et al. [33] challenge the prevailing emphasis on complex meta-learning algorithms for few-shot classification in the image domain. The authors show that learning a supervised or self-supervised representation on the meta-training set and then training a linear classifier with the learned representations by the few-shot method outperforms leading methods for few-shot classification. Their method involves combining all meta-training data into a single task to train a neural network model and using the neural network model as a fixed feature extractor during meta-testing. The findings emphasize the potential of well-learned embeddings to achieve superior few-shot classification performance across multiple benchmarks.

Laenen et al. [34] investigate the utility and efficiency of episodic training in the image domain. They question the necessity of this approach when non-parametric, metric-based methods such as episodic training prototype networks are used and do not adapt during the testing phase. They show that selecting non-episodic approaches over episodic training can lead to improved performance on several few-shot classification benchmarks. They argue

that episodic training can be an inefficient form of data use. They suggest that simpler, non-episodic methods can often be more effective.

3.2.1. Few-Shot Audio Classification

MetaAudio [1] is the first benchmark for few-shot audio classification. Authors investigate several popular approaches for few-shot learning, including MAML and Meta-Curvature [35] which are gradient-based meta-learning methods or metric-based approaches like prototypical networks, by assessing them on seven audio data sets covering sound event to audio. Overall, their experiments suggest that the gradient-based meta-learners can enhance their query performance and are more suitable than prior works (with other methods) in cases with much larger class-wise variance. They also discover a joint training mechanism across several data sets that aid better generalization to environmental sounds, as well as demystifying the challenges of domain adaptation between each data set.

Choue et al. [36] present an attentional similarity module to address the few-shot sound recognition problem. The proposed module can be used with metric-based learning methods for few-shot learning, as it enhances the ability of the model to match associated short-sound events. The authors extensively evaluate the ESC-50 and noise ESC-50 data sets and demonstrate consistent performance enhancements across five different metric-based methods.

Moummad et al. [37] contribute to the small data problem in bio-acoustic sound event detection, a domain of high importance for studying animal behavior and biodiversity. The authors, however, interestingly leverage techniques from information theory within supervised contrastive learning to effectively transfer non-redundant and diverse features across few-shot learning tasks. This consists of first pre-training a feature extractor using this regularized contrastive learning approach, and then fine-tuning it in an on-task manner with prototypical loss supervision using the nearest prototype classifier. The system is evaluated on the few-shot bio-acoustic data sets of the DCASE community, and it has achieved up to 68.19% F1-score. Kao et al. [38] combine self-supervised learning and meta-learning to

address the challenge of few-shot keyword spotting when only a few samples are available to adapt in user-defined scenarios. They use lots of models, including CPC [39], TERA [40], Wav2Vec [41], HuBERT [42], and WavLM [43] in several meta-learning algorithms such as MAML[44], Prototypical Networks, and Matching Networks. They highlight that metric-based methods such as matching networks generally outperform optimization-based methods like MAML. Additionally, they show that the encoder with fixed weights often helps get better performance, according to experiments conducted on GSC and Common Voice data sets. Heggan et al. [45] address the challenge of learning multiple inductive biases within a single model. This combines contrastive and predictive adapters with multi-task learning to train a model from scratch. The method employs an augmentation pipeline to generate correlated views of input samples to be fed to a base network equipped with lightweight neural adapters. The adapters permit task-specific parameters to be updated through either contrastive or predictive gradient updates, thereby enabling the model to store augmentation invariance and variance information. The model was evaluated on a series of few-shot audio classification tasks across 10 data sets, which encompassed both speech and non-speech data. Employing SimCLR and SimSiam as contrastive algorithms demonstrated superior performance compared to the baseline and simple multi-task approaches, giving 69% 5-way 1-shot accuracy for the ESC-50 data set.

HalluAudio [46] is a novel method for few-shot audio classification by leveraging the unique structure of audio spectrograms. The proposed method utilizes high-frequency and low-frequency parts of the spectrogram as organized concepts to enhance classification performance. It constructs high-frequency and low-frequency prototypes. It then combines them with the original spectrogram prototypes for classification. Several tests are conducted on the ESC-50 data set and a curated Kaggle18 data set. It performs better than baseline methods, giving 71.88% and 59.35% classification accuracy in a 5-way 1-shot setting on the ESC-50 and Kaggle 18, respectively. The study also compares the performance of hallucinating time-domain concepts to frequency-domain concepts. The results prove that the frequency-domain concept is superior. The results indicate that the proposed method offers an effective solution for few-shot audio classification.

Guzhov et al. establish a tri-modal CLIP [47] extension with the capacity to process text, images, and audio simultaneously. This extension draws upon the AudioSet data set for training and utilizes a range of data augmentation techniques to enhance the audio data. The training process comprises two stages: first, the audio encoder model is pre-trained on the AudioSet data set independently; second, it is jointly trained with CLIP's text and image heads. The proposed method obtains a zero-shot accuracy of 69.40% on the ESC-50 data set via an audio-only model. Elizalde et al. [48] present the CLAP model. They try to establish a connection between natural language and audio through a multi-modal space using contrastive learning. It utilizes two different pre-trained encoders for audio and text inputs. It jointly learns the similarity of these audio-text pairs without the need for labeled training data. It achieves high zero-shot performances in various data sets, such as the ESC-50, giving 82.6% accuracy. Lin et al. [49] explore the concept of leveraging multi-modal information to enhance few-shot learning for uni-modal tasks. They provide an audio-visual benchmark that shows the performance improvements for both audio and image classification, showing cross-modal training impacts positively.

4. METHODOLOGY

This section introduces the model used, the types of optimization employed, and the training schemes utilized by these optimization types. Three distinct optimizations are employed: Single-Stage Hybrid Loss Optimization (SSHLO), Single Stage Loss Optimization (SSLO), and Two Stage Loss Optimization (TSLO), via a non-parametric classifier, namely the nearest centroid classifier. SSLO is the simplest optimization that involves optimization without simCLR architecture and contrastive loss. TSLO optimization includes a simCLR-like architecture with contrastive and supervised losses in two stages. The utilized model is optimized contrastive loss and supervised loss for the first and second stages of training, respectively. SSHLO optimization modifies the SimCLR architecture and makes use of supervised loss and contrastive loss in a hybrid manner. Furthermore, the utilization of these optimizations with episodic and non-episodic training regimes is detailed. Moreover, it explains the simple feature transformations that are employed in the training and testing phases.

4.1. Audio Preprocessing

4.1.1. Mean and Variance Normalization

Mean and variance normalization is a pre-processing technique employed to scale each data point feature. To conduct this process, the mean of each feature is subtracted from the data sample. Subsequently, the result is divided by the standard deviation. This method enables enhanced performance and convergence of numerous machine learning algorithms by normalizing the data to a common scale.

4.1.2. Feature Extraction

The Mel filter bank converts audio signals into a set of perceptually relevant features. Using the Mel scale, which aligns with human pitch perception, a series of triangular filters is

applied to the power spectrum of the audio. This process extracts coefficients representing the energy in each filter.

In feature extraction for this study, the SpeechBrain [50] toolkit and its default values for the parameters [51] are used. However, some parameters are utilized differently in our study. The number of Mel-filters is 60. The left and right frames are set at 0. Moreover, sliding window length and hop length parameters are used at 25 ms and 10 ms, respectively.

4.2. Encoder

The Emphasised Channel Attention, Propagation, and Aggregation in Time Delay Neural Network (ECAPA-TDNN) model [16] has been employed as the base encoder for all methods in this thesis. The ECAPA-TDNN model is an improved X-vector architecture version that builds upon the traditional TDNN by integrating advanced techniques that significantly enhance retrieving meaningful embeddings. The SE-Res2Block is at the core of the ECAPA-TDNN. It involves the Squeeze-and-Excitation Networks (SE-Net) [52] and Res2Net [53] architectures. These play a role in recalibrating channel-wise feature maps and capturing information across a spectrum of receptive fields and scales. This allows the model to prioritize the processing of critical features over less important ones. The Attentive Statistics Pooling (ASP) layer tries to refine the representation of features, and to do that, it applies attention weights to time frames and channels. This is useful to emphasize the most informative aspects of the audio signal. Furthermore, the model aggregates features from multiple layers to benefit both shallow and deep information and make embedding more qualified.

In the proposed encoder model, there is also an MLP layer. This MLP layer comprises of two fully connected layers each containing 256 and 512 neurons with ReLU activations. It is used to enhance the encoding capability of the encoder model, as depicted in Fig. 4.1.

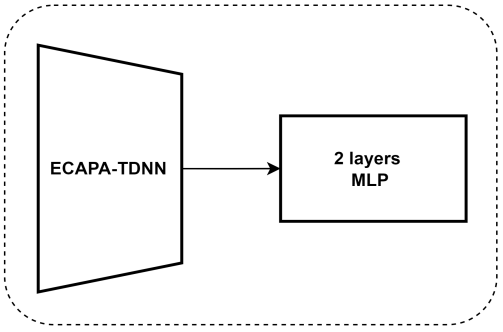

Figure 4.1 Encoder model

4.3. Data Augmentations

The data augmentation methods applied within the scope of the study are detailed in 4.1 with the parameters used and the applied values as utilized in [29]. The optimization of TSLO and SSHLO, which use simCLR and contrastive in their pipelines, leverages these augmentations.

Augmentation	**Parameter**	**Value**
White Noise	Min / Max SNR in dB 3 / 30	3/30
	Min Max f-Decay	-1/0
Mixed Noise	Min / Max SNR in dB	3/30
	Min Max f-Decay	-2/2
Pitch Shift	Min / Max Transpose Semitones	-15 / 15
Time Shift	Min / Max Shift Ratio	-0.5 / 0.5

Table 4.1 Data Augmentations

Noise Additions are frequency-based transformations. It simply means injecting random, white, or mixed-noise signals into the original samples. This transformation may assist the model in becoming more resilient against environmental noise or potential variations in recording conditions. In this study, **white noise** and **mixed noise** variations have been employed.

Pitch Shifting is a type of frequency transformation process that involves the random addition or removal of audio signals. According to the findings presented in the [29] findings,

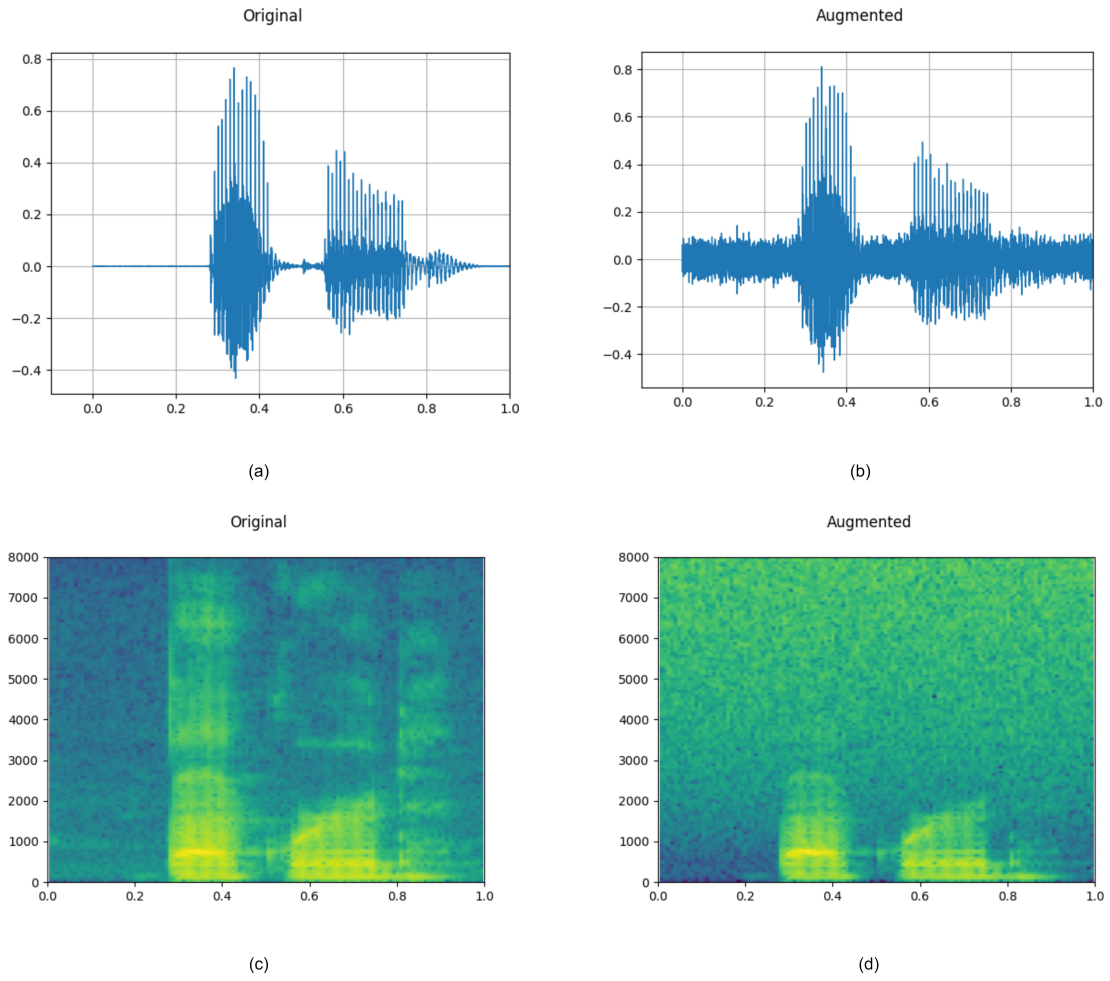

Figure 4.2 The images of representations for the Backward class from GSC data set. (a) and (b) represent the original and augmented raw waveforms, respectively. (c) and (d) represent represent original and augmented Mel-spectrograms, respectively

an allowable pitch shift range of +/- 15 pitch shifts is enough to maintain the cohesion of the input audio. This technique can simulate variations of the voice or musical notes.

Time Shifting process involves a temporal process of shifting audio signals. It is carried out by randomly rolling audio forward and backward in the time domain.

Fig. 4.2 illustrates the combined impact of all data augmentations detailed in Section 4.3. on the audio samples on the raw waveform and its Mel-spectrograms. Subfigures (a) and (c) show audio representations before applying and subfigures (b) and (d) show audio representations after applying augmentations.

4.4. Loss Functions

In this study, two different loss types are utilized: contrastive loss, and supervised loss.

4.4.1. Contrastive Loss

Normalized Temperature-scaled Cross-Entropy (NT-Xent) loss is a frequently utilized contrastive loss function in self-supervised learning tasks.

In training the SimCLR model, the data is used in pairs. These data pairs and the output of the projection layer in the simCLR model are used as representation vectors. Loss calculation is performed through the representation vectors obtained. While reducing the distance between the representation vectors of data points from the same class, it is also useful to increase the distance between the representation vectors of samples belonging to different classes.

$$\mathcal{L}_{cont} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{4}$$

In Eqn. (4), $\mathcal{L}_{cont}$ is the contrastive loss function for the positive sample pair (z_i, z_j). The numerator is the equivalent of the similarity calculated as in the Eqn. (5) between the positive pair, scaled by the temperature parameter τ. The denominator is the sum of the conjugates of the similarities of all other 2N samples in the aggregate data except z_i and itself, and this sum is also scaled by τ.

$$sim(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \tag{5}$$

4.4.2. Supervised Loss

Cross-entropy loss has been employed, which is a commonly used loss function in machine learning when working with labeled data. It is a standard choice in multi-class classification models. This loss function quantifies the degree of model fit to the actual label.
The more accurate the model's prediction, the lower the loss. In Eqn. (6), y_c is the binary value for class c of real labels. $\hat{y}_c$ represents the probability predicted by the model for class c. In this equation, C represents the total number of classes.

$$\mathcal{L}_{ce} = -\sum_{c=1}^{C} y_c \log(\hat{y}_c) \tag{6}$$

4.5. Nearest Centroid Classifier

The nearest centroid classifier is a simple and effective classification technique. It has been utilized by both episodic [9] and non-episodic [12] methods. It involves calculating the centroids of data classes and then allocating new data points to the class with the closest centroid, as given in (Eqn. 7).

$$y(\hat{x}) = \arg \min_{c \in \{1,\ldots,C\}} d(\hat{x}, \hat{x}_c) \tag{7}$$

In the context of this study, The nearest centroid classifier has been employed in both the validation and testing phases. However, it has not been utilized during the training phase. It has been utilized in two manners for experimental purposes. In the 5-shot scenario, the centroid vector is obtained by calculating the average of the feature vectors extracted by the trained encoder of the data points in the support set. while, in the 1-shot scenario, the feature vector of the instance in the support set is directly used as the centroid representation vector.

4.6. Simple Feature Transformations

In this study, feature transformations have been utilized, as in [12]. Wang et al.'s study employs two forms of basic feature transformations: L2 normalization (L2N) and centered L2 normalization (CL2N). L2N normalizes the length of each feature vector to unit length.

$$\hat{x} \leftarrow \frac{\hat{x}}{\|\hat{x}\|_2} \tag{8}$$

$$\bar{x} = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{x \in \mathcal{D}_{\text{train}}} x \tag{9}$$

In the CL2N algorithm, the first step involves calculating average of the training set from all the features as shown in the Eqn. (9). Secondly, averaged features are subtracted from test features as given in the (Eqn. 10).

$$\hat{x} \leftarrow \hat{x} - \bar{x}. \tag{10}$$

Following this, the L2 normalization is applied to the obtained features.

4.7. Optimizations

This thesis examines and contrasts three distinct optimization strategies, each of which employs the same encoder as detailed in Section 4.2. In addition, the study considers both episodic and non-episodic training methodologies in the context of each optimization strategy.

4.7.1. Single Stage Loss Optimization (SSLO)

This type of optimization is a fundamentally straightforward method. During the training phase, it only uses a supervised loss function. It is trained without the incorporation of a contrastive loss or a simCLR-like structure. During training, there is a simple single-layer linear classifier containing 512 neurons. This layer aids in the calculation of supervised loss values. The pipeline of this optimization is depicted in Fig. 4.3.

In the validation and testing phases, episodic testing is applied using the nearest centroid classifier to evaluate the obtained model. Simple feature transformations are applied to the representation vectors obtained for the test samples before feeding the classifier for both validation and testing stages.

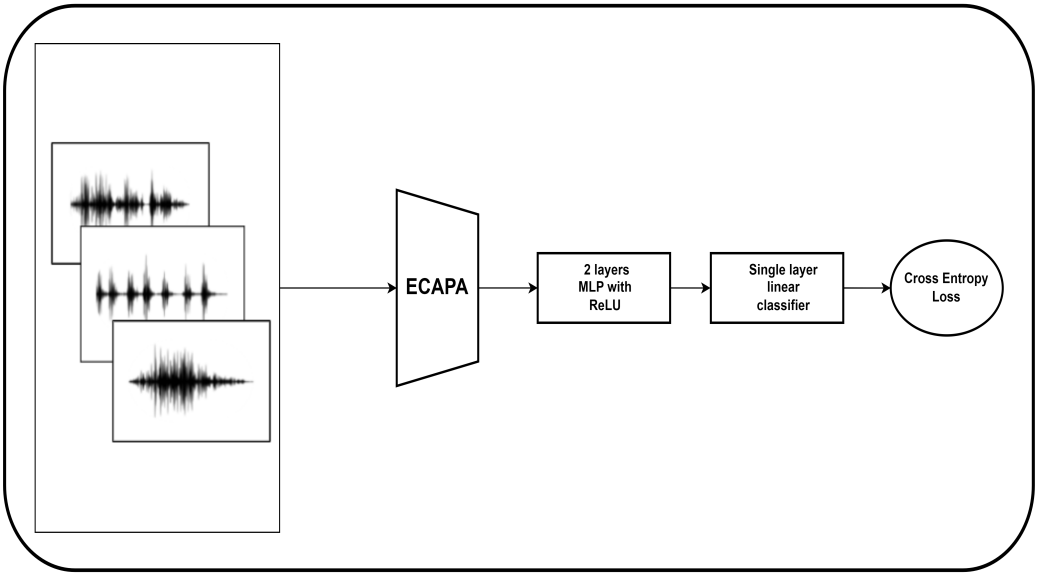

Figure 4.3 The figure depicts the pipeline for the SSLO method.

4.7.1.1. Training Procedure

There is no support set or query set in non-episodic training; conventional supervised training is conducted using a single supervised cross-entropy loss function given in (Eqn. 6) applied to all instances within a given data stream batch.

4.7.2. Two-Stage Loss Optimization (TSLO)

In TSLO optimization, there are two stages in model training. In the first stage, it uses an unsupervised contrastive loss mechanism in a simCLR-like architecture. This architecture contains a projection head module. The projection head is an MLP layer that consists of two fully connected layers, each containing 512 and 128 neurons with ReLU activations. This projection optimizes the features for contrastive losses during training. The entire encoder model (Section 4.2.) is trained using unsupervised contrastive loss, as shown in Fig. 4.4.

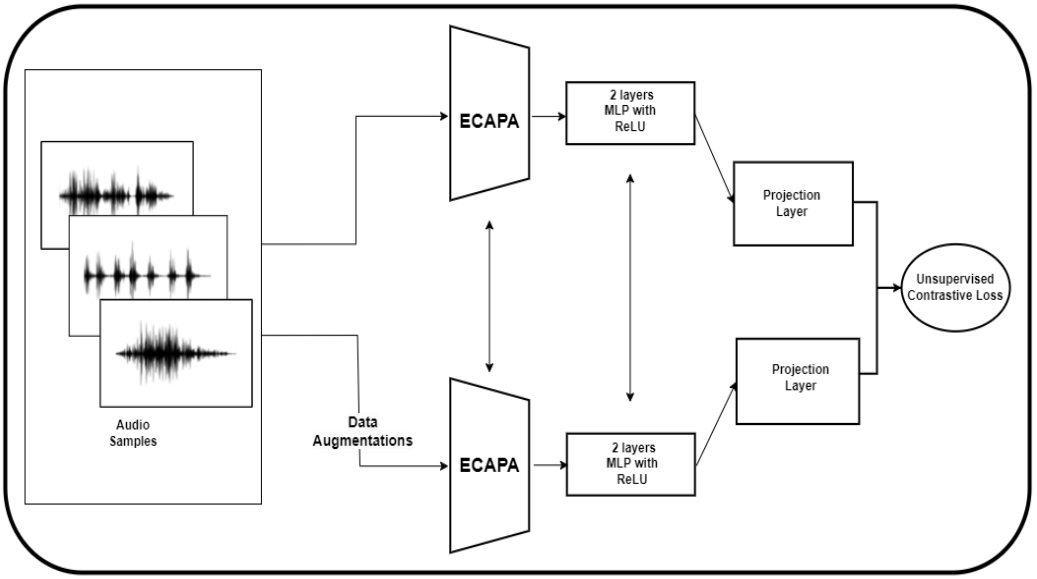

Figure 4.4 First phase of the 2-stage loss optimization

In contrast to the first phase, supervised cross entropy loss is utilized and the components used with contrastive loss are not used in the second stage. There is a single-layer linear classifier similar to that used in SSLO optimization used for supervised cross-entropy loss calculation as shown in Fig. 4.5. The ECAPA-TDNN model within the encoder model is used as a frozen entity, with its trainable parameters are closed for training. Hence, the learned parameters in the first stage are fixed and the model is used as a pre-trained model in this stage.

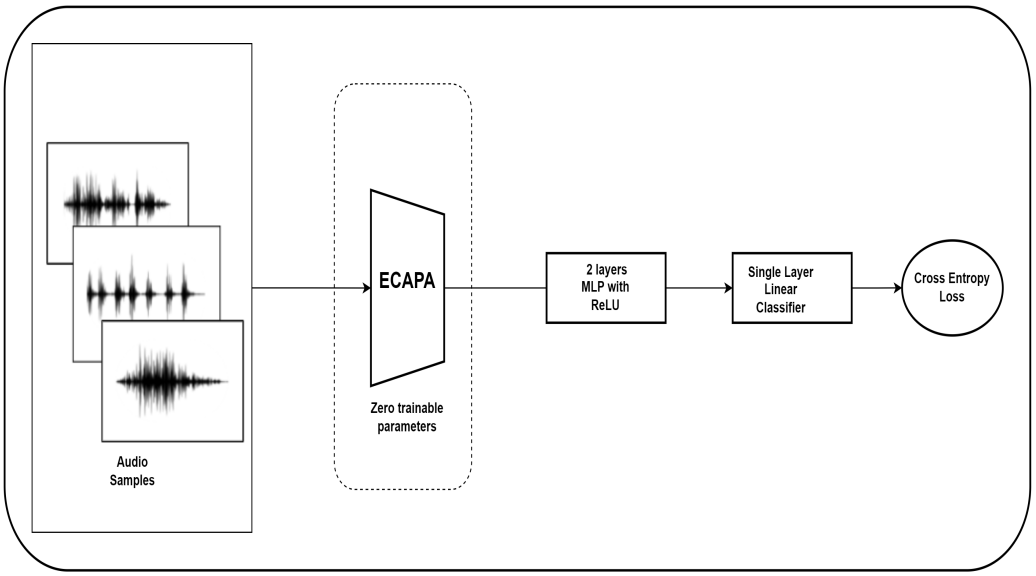

Figure 4.5 Second phase of the 2-stage model optimization

4.7.2.1. Training Procedure

In the first phase of the non-episodic training of two-stage loss optimization, audio samples are utilized without labels using only the unsupervised contrastive loss given in Section (4.4.1.). As for the second training phase, the MLP part of the encoder model is optimized using the supervised cross-entropy loss given in Eqn. (6).

4.7.3. Single Stage Hybrid Loss Optimization (SSHLO)

As opted before, the encoder model is used as detailed in Section 4.2. A projection head is present as in frameworks like SimCLR. Hence, a projection head module that is the same architecture used in TSLO optimization is appended base encoder model to transform learned representations into a space and apply contrastive learning.

An MLP layer comprises two fully connected layers, each containing 512 and 128 neurons with ReLU activations. During training, this projection head optimizes the feature space for contrastive tasks. However, it is removed during evaluation time and not used for testing purposes.

In order to benefit from supervised loss (Eqn. 6) along with contrastive loss (Eqn. 4), a single-layer linear classifier, which is also used in other optimizations, is appended after the encoder part as a branch, as can be seen in Fig. 4.6.

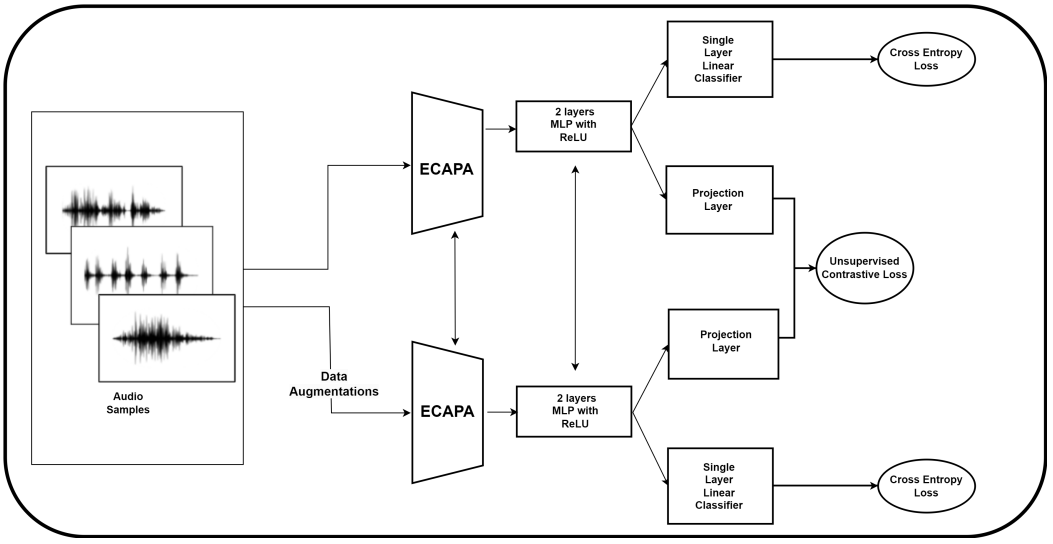

Figure 4.6 Single Stage Hybrid Model Optimization

4.7.3.1. Training Procedure

In the context of single-stage hybrid loss optimization non-episodic training, the training process does not utilize a support set or a query set as seen in episodic training. Consequently, all audio samples within a batch are duplicated, with one copy representing the original sample and the other representing the augmented version. The unsupervised contrastive loss (Eqn. 4) is calculated between these two sample sets without using their labels. Furthermore, the single-layer linear classifier enables the calculation of supervised cross-entropy losses (Eqn. 6) using the original and augmented samples in the batch. Ultimately, the losses obtained are combined as given in (Eqn. 12), and the model training is conducted in a hybrid manner.

$$\mathcal{L}_{contrastive} = \mathcal{L}_{cont}(Z_{original}, Z_{augmented}) \tag{11}$$

$$\mathcal{L}_{SSHLO} = \mathcal{L}_{contrastive} + \mathcal{L}_{CEoriginal} + \mathcal{L}_{CEaugmented} \tag{12}$$

5. EXPERIMENTAL RESULTS

5.1. Experimental Setup

5.1.1. Data Sets

In the thesis, two different and widely recognized sets of data are used, and general features are given in Table 5.1.

Data set Name	Type	Number of All Classes	Sample Length
GSCv2 (Google Speech Commands)	Spoken Command	35	1 sec
ESC-50 (Environmental Sound Classification)	Environmental	50	5 sec

Table 5.1 Data sets

5.1.1.1. Environmental Sound Classification

The ESC-50 (Environmental Sound Classification) data set [54] is a well-known data set for audio classification tasks. It originally consisted of 2000 environmental audio clips. The audio samples are separated into 40 samples per class. They were originally 44.1 kHz and down-sampled to 16 kHz for experiments.

ESC-50 is a commonly used data set for conducting benchmark audio classification studies. It includes environmental sound classes such as animal noises, natural soundscapes, human sounds, and machinery noises.

5.1.1.2. Google Speech Commands

The Google Speech Commands [55] data set is a widely used data set in audio pattern recognition tasks. It comprises two versions, the second involving 35 spoken word classes, used for experiments in this study. It provides a standardized set of 1-second-long audio utterances for training, validation, and testing.

5.1.1.3. Training Data Variations

Tables 5.2 and 5.3 illustrate the class splits that have been employed in episodic training for the ESC-50 and GSC datasets, respectively. The ESC-50 class division has been conducted according to [1]. Pre-defined 35, 5, and 10 classes out of 50 are allocated for train, validation, and test sets, respectively.

ESC-50 Class Splits		
Train	**Validation**	**Test**
35 class	**5 class**	**10 class**
clock tick, wind, pouring water, pig, fireworks	clock alarm coughing hen crackling fire breathing	airplane engine sneezing thunderstorm glass breaking cow crow church bells vacuum cleaner dog
can opening, hand saw, toilet flush, train		
sea waves, clapping, frog		
washing machine, crying baby, chainsaw		
siren, cat, sheep		
door wood knock		
car horn		
drinking sipping, helicopter, brushing teeth		
water drops, insects, snoring		
crickets, keyboard typing, rain		
door wood creaks, mouse click		
chirping birds, footsteps, rooster, laughing		

Table 5.2 ESC-50 class splits [1]

As for the GSC data set, random selection allocates approximately 70 percent of the GSC classes for the train set. This implies that we use 24 out of 35 classes for training without any overlap. The remaining classes are randomly split, and the test set involves seven distinct classes.

To reflect the real-world data scarcity problem in the experiments and to assess the efficacy of the training methodologies across different data sets, three distinct training set variants were formed for both the ESC-50 and GSC data sets, designated as SPC-5, SPC-10, and

GSCv2 Class Splits		
Train	**Validation**	**Test**
24 class	**4 class**	**7 class**
learn, up, bird nine, yes, cat off, wow, four, follow sheila, three, go eight, marvin, two, visual left, one, backward, house zero, bed, happy	five, tree no, on	down, right, forward stop, dog, six, seven

Table 5.3 GSC Class Splits

SPC-15, respectively. During the creation of these data sets, the number of samples per class in the train set was selected to be 5, 10, and 15, respectively.

Data Set	SPC-5	SPC-10	SPC-15	Validation	Test
ESC-50	175	350	525	50	150
GSCv2	120	240	360	1295	2198

Table 5.4 The sample counts for each variation in the data sets that have been applied class splitting.

In all experiments, the same data splits are employed. Table 5.4 shows the number of data samples used in the train stages in experiments for each data set.

5.1.2. Encoder Model Variations

The encoder model is composed of two principal components (Fig. 4.1): the ECAPA-TDNN backbone layer and the MLP layer. There are two different ways to use these layers in experiments. The first one is the fixed ECAPA. The ECAPA-TDNN is used as a pre-trained model, and its weights are not changed in the training stage. Only the MLP layer is trained. The second one is referred to as the adapted ECAPA. This type of usage involves training both encoder model components.

5.2. Implementation Details

To train models, the Pytorch framework [56] is employed. Implementation of the ECAPA-TDNN backbone model that is used in the encoder is obtained from HuggingFace [57]. This version is pre-trained on a large language identification data set [58] to benefit prior knowledge.

All of the trainings are conducted over 20 epochs, and the Adam optimizer [59] is used with a learning rate of 3e-4. L2 regularization weight is utilized as 1e-4. The unsupervised contrastive loss temperature value is experimentally set at 0.75.

5.3. Evaluation Process

To measure and compare our methods, N-way K-shot episodic testing is conducted. For these tests, 5-way 1-shot and 5-way 5-shot setups are prepared. A testing episode example has been shown in Fig. 5.1. Mean accuracy is measured by selecting 10,000 N-way K-shot episodes from the test set as depicted in the Alg. 7.

Algorithm 1 N-Way K-Shot Classification Evaluation

Require: $D_{\text{novel}} = \{(\tilde{x}_j, \tilde{y}_j); \tilde{x}_j \in X_{\text{novel}}, \tilde{y}_j \in Y_{\text{novel}}, j = 1, \ldots, N_{\text{novel}}\}$

Require: Number of episodes E

1: **for** e = 1,...,E **do**

2: Select randomly N classes from Y_{novel}.

3: Select randomly K samples from each class as the support set $D_S^{(e)}$.

4: Select randomly Q sample from the remaining samples of N classes as the query set $\{(\tilde{x}^{(e)}, \tilde{y}^{(e)})\}$.

5: Obtain prediction labels $\tilde{y}^{(e)} = f_{\theta^{(e)}}(D_{\text{train}}, D_S^{(e)})$.

6: Calculate accuracy

7: **end for**

Compute Avg_Acc

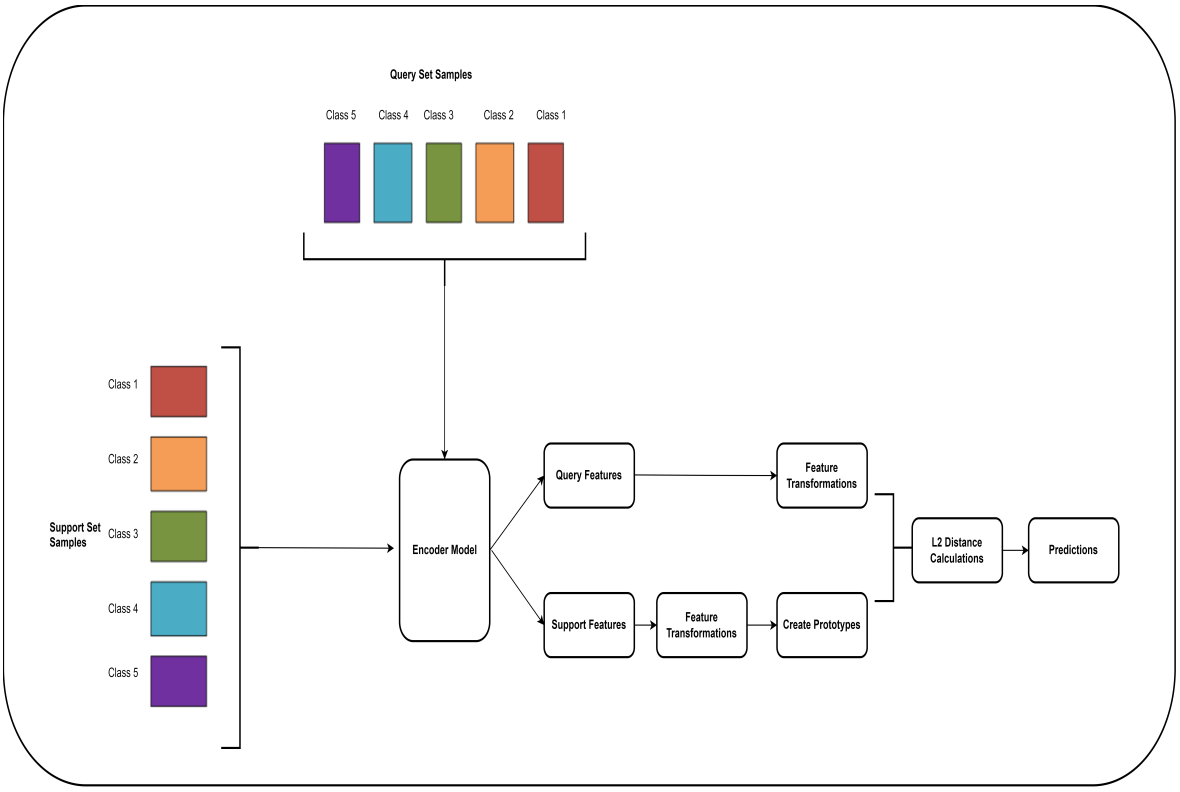

Figure 5.1 Example 5-way 1-shot test episode

5.4. Experimental Results and Discussion

Batch size values are set at 10 and 30 for 5-way 1-shot and 5-way 5-shot schemes, for non-episodic experiments, respectively. All evaluations are conducted episodically, using the 5-way 1-shot scheme.

The lack of a sufficient data sample to form an episode prevented the 5-way 5-shot training from using the SPC-5 variations. A 5-way, 5-shot episode requires the inclusion of 25 samples in the support set and 5 samples in the query set. However, the application of a random sampling operation to a data set comprising 5 samples per class results in the elimination of all samples in a class.

Table 5.5, Table 5.6, and Table 5.7 present the experimental results for SPC-5, SPC-10, and SPC-15 train data set variations for both ESC-50 and GSCv2 data sets, respectively. It can be observed that there is a tendency for the accuracy rate to increase as the number of

Data Set	ESC50		GSC	
Encoder	**Fixed ECAPA**	**Adapted ECAPA**	**Fixed ECAPA**	**Adapted ECAPA**
Non-Episodic Trainings				
TSLO	0.544 ± 0.046	0.546 ± 0.049	0.343 ± 0.044	0.374 ± 0.046
TSLO+L2N	0.541 ± 0.047	0.556 ± 0.047	0.353 ± 0.046	0.399 ± 0.045
TSLO+CL2N	0.575 ± 0.043	0.595 ± 0.046	0.359 ± 0.050	0.396 ± 0.046
SSLO	0.572 ± 0.046	0.565 ± 0.045	0.422 ± 0.046	0.481 ± 0.048
SSLO+L2N	0.597 ± 0.045	0.606 ± 0.047	$\mathbf{0.458 \pm 0.048}$	0.524 ± 0.050
SSLO+CL2N	0.596 ± 0.046	0.551 ± 0.048	0.456 ± 0.049	0.530 ± 0.046
SSHLO	0.579 ± 0.043	0.585 ± 0.045	0.425 ± 0.046	0.564 ± 0.048
SSHLO+L2N	$\mathbf{0.602 \pm 0.047}$	$\mathbf{0.622 \pm 0.045}$	0.441 ± 0.050	0.595 ± 0.049
SSHLO+CL2N	0.594 ± 0.046	0.613 ± 0.046	0.442 ± 0.048	$\mathbf{0.614 \pm 0.047}$

Table 5.5 Average accuracy scores on SPC-5 data set. Non-episodic trainings are conducted using a batch size of 10.

Data Set	ESC50		GSC	
Encoder	**Fixed ECAPA**	**Adapted ECAPA**	**Fixed ECAPA**	**Adapted ECAPA**
Non-Episodic Trainings				
TSLO	0.556 ± 0.046	0.539 ± 0.046	0.334 ± 0.044	0.400 ± 0.047
TSLO+L2N	0.567 ± 0.045	0.555 ± 0.047	0.333 ± 0.043	0.407 ± 0.046
TSLO+CL2N	0.575 ± 0.046	0.614 ± 0.045	0.354 ± 0.046	0.423 ± 0.049
SSLO	0.578 ± 0.046	0.615 ± 0.046	0.414 ± 0.047	0.528 ± 0.047
SSLO+L2N	0.604 ± 0.044	0.633 ± 0.047	$\mathbf{0.452 \pm 0.049}$	0.575 ± 0.048
SSLO+CL2N	$\mathbf{0.612 \pm 0.046}$	0.617 ± 0.046	0.447 ± 0.047	0.510 ± 0.048
SSHLO	0.589 ± 0.045	0.590 ± 0.044	0.430 ± 0.046	0.610 ± 0.048
SSHLO+L2N	0.611 ± 0.047	$\mathbf{0.644 \pm 0.045}$	0.445 ± 0.049	0.632 ± 0.047
SSHLO+CL2N	0.611 ± 0.046	0.634 ± 0.043	0.445 ± 0.050	$\mathbf{0.645 \pm 0.048}$

Table 5.6 Average accuracy scores on SPC-10 data set. Non-episodic trainings are conducted using a batch size of 10.

Data Set	ESC50		GSC	
Encoder	**Fixed ECAPA**	**Adapted ECAPA**	**Fixed ECAPA**	**Adapted ECAPA**
Non-Episodic Trainings				
TSLO	0.559 ± 0.049	0.541 ± 0.048	0.312 ± 0.043	0.421 ± 0.047
TSLO+L2N	0.566 ± 0.045	0.538 ± 0.046	0.321 ± 0.044	0.423 ± 0.048
TSLO+CL2N	0.585 ± 0.045	0.550 ± 0.047	0.333 ± 0.048	0.418 ± 0.048
SSLO	0.582 ± 0.045	0.622 ± 0.050	0.418 ± 0.045	0.539 ± 0.047
SSLO+L2N	0.598 ± 0.045	0.640 ± 0.047	$\mathbf{0.463 \pm 0.049}$	0.595 ± 0.047
SSLO+CL2N	0.600 ± 0.048	0.615 ± 0.041	0.455 ± 0.049	0.525 ± 0.048
SSHLO	0.591 ± 0.048	0.597 ± 0.049	0.427 ± 0.048	0.598 ± 0.046
SSHLO+L2N	$\mathbf{0.616 \pm 0.047}$	0.645 ± 0.044	0.445 ± 0.051	0.622 ± 0.048
SSHLO+CL2N	0.613 ± 0.042	$\mathbf{0.648 \pm 0.050}$	0.455 ± 0.046	$\mathbf{0.632 \pm 0.047}$

Table 5.7 Average accuracy scores on SPC15 data set. Non-episodic trainings are conducted using a batch size of 10.

instances contained in the classes in the training set increases. However, this increase is not proportional to the increase in the amount of data. In some cases, even slight decreases in the accuracy rate were observed.

In light of the findings presented in Table 5.6, the SSHLO with CL2N model outperforms the other optimizations on the GSC data set. Concerning Table 5.7, the highest level of accuracy is achieved by the SSHLO method with L2N transformation, giving 61.6% accuracy for fixed ECAPA. The SSHLO applying the CL2N transformation achieves even better results and gets 64.8% accuracy for not-fixed ECAPA usage on the ESC-50 SPC-15 data set. As for the GSC data set, the SSHLO with CL2N model performs the best, obtaining 63.2% accuracy and showing significant improvement when the encoder is fine-tuned compared to the fixed version. The SSHLO optimization with CL2N provides the best accuracy for the fixed ECAPA.

Data Set	ESC50		GSC	
Encoder	**Fixed ECAPA**	**Adapted ECAPA**	**Fixed ECAPA**	**Adapted ECAPA**
Non-Episodic Trainings				
TSLO	0.553 ± 0.046	0.558 ± 0.046	0.344 ± 0.045	0.393 ± 0.043
TSLO+L2N	0.569 ± 0.044	0.559 ± 0.045	0.360 ± 0.045	0.398 ± 0.045
TSLO+CL2N	0.599 ± 0.046	0.589 ± 0.043	0.375 ± 0.048	0.406 ± 0.049
SSLO	0.582 ± 0.045	0.591 ± 0.046	0.418 ± 0.046	0.506 ± 0.047
SSLO+L2N	$\mathbf{0.608 \pm 0.045}$	0.632 ± 0.046	0.450 ± 0.047	0.588 ± 0.048
SSLO+CL2N	0.604 ± 0.047	0.627 ± 0.046	0.452 ± 0.051	0.556 ± 0.048
SSHLO	0.584 ± 0.044	0.630 ± 0.045	0.440 ± 0.046	0.609 ± 0.047
SSHLO+L2N	0.605 ± 0.047	$\mathbf{0.658 \pm 0.046}$	$\mathbf{0.455 \pm 0.051}$	$\mathbf{0.634 \pm 0.048}$
SSHLO+CL2N	0.606 ± 0.046	0.639 ± 0.045	$\mathbf{0.455 \pm 0.049}$	0.633 ± 0.048

Table 5.8 Average accuracy scores on SPC-10 data set variations. Non-episodic trainings are conducted using a batch size of 30.

Data Set	ESC50		GSC	
Encoder	**Fixed ECAPA**	**Adapted ECAPA**	**Fixed ECAPA**	**Adapted ECAPA**
Non-Episodic Trainings				
TSLO	0.573 ± 0.044	0.557 ± 0.048	0.342 ± 0.041	0.409 ± 0.044
TSLO+L2N	0.584 ± 0.046	0.554 ± 0.046	0.351 ± 0.048	0.396 ± 0.048
TSLO+CL2N	0.600 ± 0.042	0.587 ± 0.043	0.372 ± 0.048	0.418 ± 0.048
SSLO	0.587 ± 0.048	0.612 ± 0.046	0.398 ± 0.047	0.513 ± 0.048
SSLO+L2N	0.604 ± 0.047	0.626 ± 0.045	0.444 ± 0.048	0.586 ± 0.048
SSLO+CL2N	0.595 ± 0.046	0.636 ± 0.045	0.459 ± 0.046	0.559 ± 0.047
SSHLO	0.592 ± 0.045	0.614 ± 0.045	0.435 ± 0.048	0.623 ± 0.047
SSHLO+L2N	0.614 ± 0.046	0.646 ± 0.044	0.450 ± 0.050	0.644 ± 0.045
SSHLO+CL2N	$\mathbf{0.616 \pm 0.045}$	$\mathbf{0.655 \pm 0.047}$	$\mathbf{0.457 \pm 0.047}$	$\mathbf{0.651 \pm 0.049}$

Table 5.9 Average accuracy scores on SPC-15 data set. Non-episodic trainings are conducted using a batch size of 30.

Table 5.8 and Table 5.9 present the experimental results for the SPC-10 and SPC-15 train data set variations of the ESC-50 and GSCv2 data sets, respectively. In these experiments, episodic training is conducted with a 5-way 5-shot scheme, while non-episodic training is applied via a batch size of 30.

When Table 5.8 and Table 5.9 are considered together, it can be seen that non-episodic training benefits from the increase in batch size. The SSHLO optimization has demonstrated efficacy when compared to other optimizations. In the majority of instances, it yields the most favorable outcomes in Table 5.8 and Table 5.9. In a single instance, the results achieved by SSLO optimization were superior, with a margin of 0.2%.

Data Set	ESC-50 (SPC-5)	GSC (SPC-5)	ESC-50 (SPC-10)	GSC (SPC-10)	ESC-50 (SPC-15)	GSC (SPC-15)
Non-Episodic Trainings						
TSLO	0.546	0.374	0.539	0.400	0.541	0.421
TSLO+L2N	0.556	0.399	0.555	0.407	0.538	0.423
TSLO+CL2N	0.595	0.396	0.614	0.423	0.550	0.419
SSLO	0.565	0.481	0.615	0.528	0.622	0.539
SSLO+L2N	0.606	0.524	0.633	0.575	0.640	0.595
SSLO+CL2N	0.551	0.530	0.617	0.510	0.615	0.525
SSHLO	0.585	0.564	0.590	0.610	0.597	0.598
SSHLO+L2N	**0.622**	0.595	**0.644**	0.632	0.645	0.622
SSHLO+CL2N	0.613	**0.614**	0.634	**0.644**	**0.648**	**0.632**
Episodic Trainings						
ProtoNets	0.618	0.471	0.615	0.462	0.626	0.479

Table 5.10 Average accuracy scores obtained with adapted ECAPA. The episodic trainings are conducted via a 5-way 1-shot scheme. Non-episodic trainings are conducted using a batch size of 10.

Tables 5.10 and 5.11 present average accuracy scores obtained using the adapted ECAPA model for all training data variations on both the ESC-50 and GSC data sets. The accuracy results are added and obtained using the Prototypical Networks model (Section 2.4.1.1.) for the episodic training. To make a fair comparison, episodic training is applied with the same encoder architecture on the same data splits. Furthermore, the number of epochs is used as equal to 20 used in non-episodic trainings. Episode count in one epoch is adjusted with the corresponding batch size in non-episodic training.

Data Set	ESC-50 (SPC-10)	GSC (SPC-10)	ESC-50 (SPC-15)	GSC (SPC-15)
Non-Episodic Trainings				
TSLO	0.558	0.393	0.557	0.409
TSLO+L2N	0.559	0.398	0.554	0.396
TSLO+CL2N	0.589	0.406	0.587	0.418
SSLO	0.591	0.506	0.612	0.513
SSLO+L2N	0.632	0.588	0.626	0.586
SSLO+CL2N	0.627	0.556	0.636	0.559
SSHLO	0.630	0.609	0.614	0.623
SSHLO+L2N	**0.658**	**0.634**	0.646	0.644
SSHLO+CL2N	0.639	0.633	**0.655**	**0.651**
Episodic Trainings				
ProtoNets	0.590	0.478	0.546	0.507

Table 5.11 Average accuracy scores obtained with adapted ECAPA. The episodic trainings are conducted via a 5-way 5-shot scheme. Non-episodic trainings are conducted using a batch size of 30.

Considering the results of Table 5.10 and Table 5.11 we can summarize our observations as follows:

- **Optimization Techniques**

  The proposed SSHLO method consistently demonstrates the highest accuracy scores. Only a few cases exhibiting narrow margins of superiority are surpassed by it. Therefore, it can be concluded that hybrid optimization of losses represents the optimum optimization strategy across the entire data set variations on the ESC-50 and GSC data sets when applying the non-episodic training approach.

  The SSLO method can learn task-specific representations with its simplicity and surpass the TSLO method. The SSHLO method utilizes both types of losses in a more intricate manner than the SSLO optimization, and it appears to learn more robust and generalizable representations, outperforming the SSLO method in most cases.

Observations show that the TSLO method generally yields poorer results than other methods. There may be several reasons for this poor performance. When training with the comparative loss function, the data augmentation methods applied to the audio samples may have been insufficient. Since the first stage of the TSLO training is done without using the supervised loss function, the importance of the data augmentation methods is greater than that of the SSHLO optimization. Furthermore, in the secondary training phase, the ECAPA-TDNN model is closed for adaptation following the initial unsupervised training stage. Consequently, the model may be unable to adapt to the representations acquired during the first unsupervised training phase, which may not be optimally suited to a given classification task.

- **Comparison to ProtoNets**

  The SSHLO and the SSLO optimizations method provide superior results compared to ProtoNets results. As stated before, ProtoNets utilize an episodic training approach whereas other optimization methods use non-episodic training. Given that a pre-trained model is used for class-balanced data sets, it can be said that non-episodic training is a better option than episodic training despite its complexity. ProtoNets outperforms the TSLO method. Regarding these results, it has been proven that the TSLO optimization approach is the worst approach when the amount of training data is low, as it is used in this study.

- **Impact of ECAPA Model Adaptation**

  The adapted utilization of the ECAPA model, part of the encoder model, has been demonstrated to yield positive effects on optimizations for all data set variations on both the ESC-50 and GSCv2 data sets. In comparison to a fixed approach, the results indicate that this method produces significant performance gains in almost every experiment.

  In the fixed-weight model, the encoder is pre-trained, and its parameters remain unchanged during training. This approach is intended to reduce computation and avoid

overfitting. However, this approach became problematic due to the variations in the data sets we used. The scarcity of samples makes the learning process difficult. As for the adapted use of the ECAPA-TDNN model, it allows representation learning that is more generalizable and stable than the fixed approach.

- **Effect of Simple Feature Transformations**
  Analysis of the results revealed that applying transformations consistently led to the highest scores. This serves as compelling evidence that simple feature transformations enhance performance. Observations show that the CL2N transformation exhibits greater efficacy than the L2N transformation, but the observed difference is often minimal, and subtle distinctions often determine the superior transformation. Centering process in CL2N transformation makes normalization more effective for classification tasks.

5.4.1. Ablation Study

The SSHLO optimization method has been demonstrated to be effective for a limited number of audio classifications, as outlined in Section 5.4.. As previously stated in Section 4.7.3., two distinct training methodologies are employed, each with a different loss function utilization. This thesis evaluates the impact of these hybrid approaches on the SPC-15 training dataset variation for both datasets.

As Table 5.12 indicates, CE and CL represent the total loss components employed in SSHLO episodic training. These are calculated using supervised cross-entropy and unsupervised contrastive losses, respectively. (Section 4.7.3.1.)

5.4.1.1. Loss Effects in Non-Episodic Training

Table 5.12 indicates that cross-entropy loss is more effective than contrastive loss for both data sets. Indeed, an examination of the results obtained from the GSC dataset reveals that the

	ESC-SPC15	GSC-SPC15
SSHLO	$\mathbf{0.596 \pm 0.045}$	$\mathbf{0.603 \pm 0.047}$
w/o CE	0.531 ± 0.048	0.535 ± 0.049
w/o CL	0.575 ± 0.046	0.600 ± 0.046

Table 5.12 Average accuracy scores on SPC-15 data set. Non-episodic trains are conducted using a batch size of 10.

contrastive loss contribution is relatively insignificant. However, the application of combined losses yields the highest accuracy values.

6. CONCLUSION

In this thesis, an audio classification has been applied using few-shot learning for cases with a limited amount of data. Extensive experiments have been carried out with three different types of optimization models. In these experiments, two different data sets have been utilized, which are ESC-50, which contains environmental non-speech sounds, and GSC, which includes simple spoken commands. Three data set variations—5, 10, and 15 samples per class have been created for each data set. The optimization methodologies are evaluated by the 5-way 1-shot few-shot classification scheme. The effects of simple feature transformations have been observed during the evaluations.

When the collected findings are examined, it can be seen that the proposed single-stage hybrid loss optimization (SSHLO) method performs better than other optimization methods in terms of classification accuracy. The SSHLO method yielded the most optimal results among all the models employed for all variations of both datasets. On the SPC-5 training set variation, it achieves 62.2% with L2 normalization for the ESC-50 dataset and 61.4% with CL2N transformation for the GSC dataset. With the SPC-10 training data set variation, the SSHLO method achieves 65.8% with L2N transformation on the ESC-50 data set and 64.4% with CL2N transformation on the GSC data set. The SSHLO with CL2N transformation yields the best scores for the SPC-15 training dataset, with the SSHLO model giving scores of 65.5% and 65.1%, respectively.

The episodic training approach stands out as an important place in the few-shot learning area. For this reason, comparisons have been conducted with Prototypical Networks which is a well-known few-shot learning model utilizing the episodic training scheme. Upon analyzing the utilized data sets and the obtained outcomes from the experimental setups, the SSHLO optimization model gives higher accuracy scores than Prototypical Networks when they are utilized with the same large-scale pre-trained encoder model.

6.1. Future Works

This thesis presents the results obtained by using a single backbone model. The ECAPA-TDNN model has been selected as the encoder model backbone due to considerations regarding training and inference time, as well as the model's success. Furthermore, the comparison between episodic and non-episodic training approaches is limited.

To enhance the quality of the study's findings and facilitate the development of other potential lines of research, the following future studies can be conducted:

- It is possible to observe changes in the experimental classification performance of optimizations using various backbone models. To illustrate, the X-vector [15] and CNN14 [60] models may be good candidates considered more lightweight than the model used in this thesis. In cases where the hardware is sufficient, backbone models such as Whisper [61], Hubert [42], Wav2vec [41], and AST [62], which are pre-trained with very large data sets, can be employed.
- A more comprehensive analysis of the differences between episodic and non-episodic training approaches is required. A comparison with known methods such as MatchingNet [10] methods utilizing episodic training, might facilitate a more accurate evaluation of the results. Furthermore, the implementation of episodic training schemes in the optimization models employed in this thesis could facilitate the generation of more reliable results in comparing episodic and non-episodic training experiments. Due to time constraints, we did not explore these models in this study.

REFERENCES

[1] Calum Heggan, Sam Budgett, Timothy M. Hospedales, and Mehrdad Yaghoobi. Metaaudio: A few-shot audio classification benchmark. In *International Conference on Artificial Neural Networks*. **2022**.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, **2020**.

[3] Enes Furkan Çiğdem and Hacer Yalim Keleş. Few-shot audio classification using contrastive training. In *2024 32nd Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. **2024**. doi:10.1109/SIU61531.2024.10600788.

[4] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, **2015**.

[5] Yuan Gong, Cheng-I Lai, Yu-An Chung, and James R. Glass. Ssast: Self-supervised audio spectrogram transformer. *ArXiv*, abs/2110.09784, **2021**.

[6] Behnaz Bahmei, Elina Birmingham, and Siamak Arzanpour. Cnn-rnn and data augmentation using deep convolutional generative adversarial network for environmental sound classification. *IEEE Signal Processing Letters*, 29:682–686, **2022**.

[7] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 646–650, **2022**.

[8] Nik Vaessen and David A. van Leeuwen. Fine-tuning wav2vec2 for speaker recognition. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7967–7971, **2021**.

[9] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175, **2017**.

[10] Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Neural Information Processing Systems*. **2016**.

[11] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, **2017**.

[12] Yan Wang, Wei-Lun Chao, Kilian Q. Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *ArXiv*, abs/1911.04623, **2019**.

[13] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*. **2013**.

[14] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio López-Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056, **2014**.

[15] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333. **2018**. doi:10.1109/ICASSP.2018.8461375.

[16] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *Interspeech 2020*, **2020**. doi:10.21437/interspeech.2020-2650.

[17] Z. Zhao, Zhuo Li, Wenchao Wang, and Pengyuan Zhang. Pcf: Ecapa-tdnn with progressive channel fusion for speaker verification. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, **2023**.

[18] Nauman Dawalatabad, Mirco Ravanelli, Franccois Grondin, Jenthe Thienpondt, Brecht Desplanques, and Hwidong Na. Ecapa-tdnn embeddings for speaker diarization. In *Interspeech*. **2021**.

[19] Jinlong Xue, Yayue Deng, Yichen Han, Ya Li, Jianqing Sun, and Jiaen Liang. Ecapa-tdnn for multi-speaker text-to-speech synthesis. *2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 230–234, **2022**.

[20] R. Pappagari, Tianzi Wang, Jesús Villalba, Nanxin Chen, and Najim Dehak. X-vectors meet emotions: A study on dependencies between emotion and speaker recognition. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7169–7173, **2020**.

[21] Spandan Dey, Md Sahidullah, and Goutam Saha. Cross-corpora spoken language identification with domain diversification and generalization. *Computer Speech & Language*, 81:101489, **2023**.

[22] David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. Spoken language recognition using x-vectors. In *The Speaker and Language Recognition Workshop*. **2018**.

[23] Linus Ericsson, Henry G. R. Gouk, Chen Change Loy, and Timothy M. Hospedales. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39:42–62, **2021**.

[24] Shuo Liu, Adria Mallol-Ragolta, Emilia Parada-Cabeleiro, Kun Qian, Xingshuo Jing, Alexander Kathan, Bin Hu, and Björn Schuller. Audio self-supervised learning: A survey. *Patterns*, 3, **2022**.

[25] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758. **2021**.

[26] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *ArXiv*, abs/2103.03230, **2021**.

[27] Jonah Anton, Harry Coppock, Pancham Shukla, and Björn Schuller. Audio barlow twins: Self-supervised audio representation learning. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, **2022**.

[28] Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive learning of general-purpose audio representations. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3875–3879, **2020**.

[29] Haider Al-Tahan and Yalda Mohsenzadeh. Clar: Contrastive learning of auditory representations. In *International Conference on Artificial Intelligence and Statistics*, pages 2530–2538. PMLR, **2021**.

[30] Alireza Nasiri and Jianjun Hu. Soundclr: Contrastive learning of representations for improved environmental sound classification. *CoRR*, abs/2103.01929, **2021**.

[31] Jit Yan Lim, Kian Ming Lim, Chin Poo Lee, and Yong Xuan Tan. Scl: Self-supervised contrastive learning for few-shot image classification. *Neural networks : the official journal of the International Neural Network Society*, 165:19–30, **2023**.

[32] Jit Yan Lim, Kian Ming Lim, Chin Poo Lee, and Yong Xuan Tan. Ssl-protonet: Self-supervised learning prototypical networks for few-shot learning. *Expert Syst. Appl.*, 238:122173, **2023**.

[33] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *European Conference on Computer Vision*. **2020**.

[34] Steinar Laenen and Luca Bertinetto. On episodes, prototypical networks, and few-shot learning. In *Neural Information Processing Systems*. **2020**.

[35] Eunbyung Park and Junier B. Oliva. Meta-curvature. *ArXiv*, abs/1902.03356, **2019**.

[36] Szu-Yu Chou, Kai-Hsiang Cheng, Jyh-Shing Roger Jang, and Yi-Hsuan Yang. Learning to match transient sound events using attentional similarity for few-shot sound recognition. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 26–30, **2018**.

[37] Ilyass Moummad, Romain Serizel, and Nicolas Farrugia. Regularized contrastive pre-training for few-shot bioacoustic sound detection, **2024**.

[38] Wei-Tsung Kao, Yue Wu, Chia-Ping Chen, Zhi-Sheng Chen, Yu-Pao Tsai, and Hung yi Lee. On the efficiency of integrating self-supervised learning and meta-learning for user-defined few-shot keyword spotting. *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 414–421, **2022**.

[39] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, **2018**.

[40] Andy T. Liu, Shang-Wen Li, and Hung yi Lee. Tera: Self-supervised learning of transformer encoder representation for speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2351–2366, **2020**.

[41] Alexei Baevski, Henry Zhou, Abdel rahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *ArXiv*, abs/2006.11477, **2020**.

[42] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdel rahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, **2021**.

[43] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Micheal Zeng, and Furu Wei. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16:1505–1518, **2021**.

[44] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400, **2017**.

[45] Calum Heggan, Tim Hospedales, Sam Budgett, and Mehrdad Yaghoobi. Mt-slvr: Multi-task self-supervised learning for transformation in (variant) representations. *arXiv preprint arXiv:2305.17191*, **2023**.

[46] Zhongjie Yu, Shuyang Wang, Lin Chen, and Zhongwei Cheng. Halluaudio: Hallucinate frequency as concepts for few-shot audio classification. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. **2023**. doi:10.1109/ICASSP49357.2023.10095663.

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. **2021**.

[48] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap learning audio concepts from natural language supervision. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, **2023**.

[49] Zhiqiu Lin, Samuel Yu, Zhiyi Kuang, Deepak Pathak, and Deva Ramana. Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19325–19337, **2023**.

[50] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. Speechbrain: A general-purpose speech toolkit, **2021**.

[51] Feature extraction module. `https://speechbrain.readthedocs.io/en/latest/API/speechbrain.lobes.features.html#speechbrain.lobes.features.Fbank`. Last Access: 2024-08-22.

[52] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, **2017**.

[53] Shanghua Gao, Ming-Ming Cheng, Kai Zhao, Xinyu Zhang, Ming-Hsuan Yang, and Philip H. S. Torr. Res2net: A new multi-scale backbone architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:652–662, **2019**.

[54] Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018. **2015**.

[55] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition, **2018**.

[56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., **2019**.

[57] Pretrained ecapa-tdnn model. `https://huggingface.co/speechbrain/lang-id-voxlingua107-ecapa`. Last Access: 2024-07-21.

[58] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*. **2018**.

[59] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, **2014**.

[60] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, **2019**.

[61] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. *ArXiv*, abs/2212.04356, **2022**.

[62] Yuan Gong, Yu-An Chung, and James R. Glass. Ast: Audio spectrogram transformer. *ArXiv*, abs/2104.01778, **2021**.