UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
Departamento de Estatística e Informática - DEINFO

PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

Speaker Identification Using Audio Looping and Margin-Based Loss Functions in a Modified VGG16 Framework

Elliot Quinten Crispiniano Garcia

Dissertação de Mestrado

RECIFE - PE
2025

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
Departamento de Estatística e Informática - DEINFO

Elliot Quinten Crispiniano Garcia

Speaker Identification Using Audio Looping and Margin-Based Loss Functions in a Modified VGG16 Framework

Trabalho apresentado ao Programa de PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA do Departamento de Estatística e Informática - DEINFO da UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO como requisito parcial para obtenção do grau de Mestre em INFORMÁTICA APLICADA.

Orientador: *Prof. Dr. Tiago Alessandro Espínola Ferreira*

RECIFE - PE
2025

Dados Internacionais de Catalogação na Publicação
Sistema Integrado de Bibliotecas da UFRPE
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

G216s Garcia, Elliot Quinten Crispiniano.
Speaker Identification Using Audio Looping and Margin-Based Loss Functions in a Modified VGG16 Framework / Elliot Quinten Crispiniano Garcia. – Recife, 2025.
56 f.; il.

Orientador(a): Tiago Alessandro Espínola Ferreira.

Dissertação (Mestrado) – Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Biometria e Estatística Aplicada, Recife, BR-PE, 2025.

Inclui referências.

1. Identificação do locutor. 2. Aumento de dados. 3. Espectrograma Mel. 4. VoxCeleb1 I. Ferreira, Tiago Alessandro Espínola, orient. II. Título

CDD 519.5

Acknowledgements

I'd like to express my thanks to Prof. Dr. Tiago Ferreira for his seemingly never-ending patience, understanding, and kindness during the process of writing my dissertation.

My father, for incentivizing me to start doing this masters, and his never-ending advice and words of encouragement, always ready to think outside the box and lift me up. You are my source of inspiration in life and I can only strive to be like you.

My mother (who I affectionally call mumsypoffs), despite not always knowing what is going on, she is always ready to give me a hug and make my day just that little bit better.

To my brothers, Ilya and Hendrik, for their unwavering support and endless jokes, you guys never fail at making me crack a smile.

To my colleagues, especially Adriana Falcão, I really appreciate all you have done for me and my family. I am proud to call you my friend.

My friends, who at times reminded me that coffee is meant to be enjoyed, and still try to convince me that perhaps I shouldn't drink liters of the stuff at 2 in the morning.

And last but certainly not least,

To everyone in the research group I am part of, IFROG, for all the cooperation and learning.

I couldn't have done it without you.

If we knew what it was we were doing, it would not be called research, would it?

—ALBERT EINSTEIN

Abstract

This dissertation evaluates the performance of two margin-based loss functions—ArcFace and CosFace—against traditional Softmax in text-independent, closed-set speaker identification. Using the VoxCeleb1 dataset, a modified VGG16 architecture was implemented with global average pooling, a custom 256-dimensional L2-normalized embedding layer, and a classification head. To address short utterance limitations, an audio looping augmentation technique extended recordings to fixed durations while preserving spectral characteristics. Mel-spectrograms were generated for three-second and ten-second clips at three resolutions. Results show that ten-second clips consistently outperformed three-second clips, with accuracy gains ranging between 7.75%–15.17%. The $432 \times 288 \times 3$ configuration yielded the best performance across most settings. CosFace achieved the highest accuracy (83.15%), outperforming ArcFace (80.79%) and Softmax (76.41%). While results align with existing literature, this study provides a reproducible comparison of loss functions, identifies effective input dimensions. Furthermore it validates a simple augmentation strategy for limited-duration audio.

Keywords: Speaker Identification, CosFace, ArcFace, Softmax, VGG16, Data Augmentation, Mel-Spectrogram, VoxCeleb1

Resumo

Esta dissertação avalia o desempenho de duas funções de perda baseadas em margem—ArcFace e CosFace—em comparação com o Softmax tradicional na identificação de locutor independente de texto e conjunto fechado. Utilizando o conjunto de dados VoxCeleb1, foi implementada uma arquitetura VGG16 modificada com pooling médio global, uma camada de embedding personalizada de 256 dimensões normalizada por L2, e uma cabeça de classificação. Para abordar as limitações de falas curtas, uma técnica de aumento de dados por repetição de áudio estendeu as gravações para durações fixas preservando as características espectrais. Mel-espectrogramas foram gerados para clipes de três segundos e dez segundos em três resoluções. Os resultados mostram que clipes de dez segundos consistentemente superaram clipes de três segundos, com ganhos de precisão variando entre 7,75%–15,17%. A configuração $432 \times 288 \times 3$ produziu o melhor desempenho na maioria das configurações. CosFace alcançou a maior precisão (83,15%), superando ArcFace (80,79%) e Softmax (76,41%).

Embora os resultados estejam alinhados com a literatura existente, este estudo fornece uma comparação reproduzível de funções de perda, identifica dimensões de entrada eficazes, e valida uma estratégia simples de aumento de dados para áudio de duração limitada.

Palavras-chave: Identificação de Locutor, CosFace, ArcFace, Softmax, VGG16, Aumento de Dados, Mel-espectrograma, VoxCeleb1

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Introduction

Speaker Recognition (SR) has become a widely used application for security systems, virtual assistants, and personalized user experiences [AWB+21, M+21]. Spectrograms provide a time-frequency representation of speech signals, capturing both temporal and spectral information for distinguishing speakers [YR18, ASSL19, ATL19]. These spectrograms are typically processed by Convolutional Neural Networks (CNNs) trained in a supervised manner on labeled datasets. However, obtaining sufficient high-quality training data, especially for shorter utterance lengths, presents a significant challenge. To overcome this, data augmentation techniques are used to enrich datasets, creating more robust training samples [NCZ17, ATL19, YR18]. This research specifically employs audio looping to extend shorter recordings, thereby ensuring consistent and informative inputs for our CNN models.

Furthermore, to maximize the potential of these enhanced data representations, we focus on improving the learning process itself. While prevailing classification loss functions, such as Softmax Loss, have shown success, they do not explicitly optimize feature embeddings to enforce higher intra-class similarity and inter-class diversity. To remedy these limitations, this study investigates the comparative effectiveness of two advanced loss functions, namely ArcFace and CosFace [DGXZ19, WWZ+18], against the traditional Softmax loss [NCZ17], evaluating their performance in conjunction with our data augmentation strategy.

1.1.1 Objectives

This work has the general objective:

> Propose a novel data augmentation approach for short audios for speaker identification based on spectral analysis of audio and image recognition using deep neural networks.

To achieve this general objective, three specific objectives are presented,

- **Evaluate Advanced Loss Functions**: To assess the comparative effectiveness of ArcFace and CosFace loss functions against traditional Softmax loss for Speaker Identification (SI), using our modified VGG16 architecture with mel-spectrogram image inputs.
- **Enhance Feature Representation via Data Augmentation**: To investigate the impact of audio sample duration on feature representation and to develop a novel audio looping

technique for extending shorter audio samples (3-second) to longer lengths (10-second). This augmentation aims to significantly improve SI accuracy by providing richer temporal information.

- **Optimize Input Configurations**: To systematically analyze the effect of varying mel-spectrogram image dimensions ($224 \times 224 \times 3$, $448 \times 448 \times 3$, and $432 \times 288 \times 3$) on model performance, identifying optimal configurations that balance computational efficiency with identification accuracy.

1.1.2 Main Contributions

The primary contributions of this work are:

- **Empirical Validation of Advanced Loss Functions**: We demonstrate that ArcFace and CosFace loss functions yield superior speaker identification accuracy compared to traditional Softmax loss when used with our modified VGG16 architecture and mel-spectrogram inputs.
- **Demonstration of Audio Looping Efficacy**: Our audio looping data augmentation technique significantly boosts identification accuracy by providing more robust temporal information from shorter audio samples, proving effective in overcoming data duration limitations.
- **Identification of Optimal Spectrogram Dimensions**: We identify specific mel-spectrogram dimensions ($432 \times 288 \times 3$) that offer a balance between computational cost and speaker identification performance.

1.1.3 Outline

The document is organized as follows.

- Chapter 2 presents related works on SI systems, establishing the context for our contributions.
- Chapter 3 presents important concepts and definitions to set the groundwork for the Methodology section.
- Chapter 4 describes our methodology, including preprocessing techniques applied to the VoxCeleb1 dataset[1] [NCZ17], the advanced loss functions employed and their theoretical advantages, and the architectural modifications made to the VGG16 model.
- Chapter 5 presents experimental results and discusses the impact of audio length and looping techniques, the comparative effectiveness of different loss functions, the effects of varying mel-spectrogram dimensions on identification accuracy, and observations made during the comparison with other studies.

[1] https://www.robots.ox.ac.uk/~vgg/data/voxceleb/vox1.html

- Chapter 6 concludes the paper with a summary and a reflection of findings and implications for future research.

CHAPTER 2

Related Work

2.1 Related Work

This section is to briefly review several related or similar works in line with this paper.

As presented in the general objective, this work approaches two branches. The first to characterize the audio information, with a spectral image of the sounds, and the second to use deep neural networks to deal with the spectral images. The technique of generating speech-based spectrograms is implemented during the feature extraction stage, while convolutional neural networks are used during the classification stage.

Nagrani *et al.* [NCZ17] originally introduced and tested the VoxCeleb1 dataset. They used random sampling to randomly select three second segments from each speech and generated mel-spectrogram images of size $512 \times 300 \times 3$. For classification, they used a modified VGG CNN, having changed the final maxpool layer to avgpool. They achieved a top-1 accuracy of 80.5% with the traditional Softmax loss function.

Anand *et al.* [ASSL19] used random sampling to make 3-second random selections of each speech and then convert them to spectrogram images of size $128 \times 300 \times 1$. In the classification stage, the spectrograms are classified by different CNNs, such as VGG, ResNet, and CapsuleNet. They employed a combination of Margin Loss for training the capsule networks and Prototypical Loss for generalizing under unseen speakers. The ResNet classifier for 200 classes from the Voxceleb1 dataset has the best result of top-1 = 71.8%.

An *et al.* [ATL19] improved upon the original SI system's accuracy [NCZ17] by altering the VGG network, specifically replacing the final max pooling layer with an average pooling layer and adding a self-attention layer before this pooling layer. This modification enabled the system to manage variable-length segments effectively. The study employed a combination of a penalization term alongside the traditional cross-entropy loss. For feature extraction, the researchers utilized mel-spectrograms, extracted from three-second audio segments, with dimensions of $512 \times 300 \times 3$. The resulting accuracy on the VoxCeleb1 dataset was a top-1 of 90.8%.

Similarly, Yadav and Rai [YR18] extracted mel-spectrograms from three-second utterances using random sampling; this time, however, they applied noise injection for data augmentation. The mel-spectrograms were generated with size $301 \times 161 \times 1$. One of the methods used in the classification stage was a VGG13-based CNN by adding a batch normalization layer after every convolutional layer. The system employs joint supervision of Softmax and Center Loss, achieving a top-1 accuracy of 89.5% on the VoxCeleb1 dataset.

Sharif *et al.* [SNRM25] augmented their data using a variety of methods, beginning with random sampling and then including several image-based techniques designed to preserve the

nature of mel-spectrogram images while increasing training samples. The flip horizontal (FH) method rotated images around the time axis without affecting frequency content, while the time shift (TS) shifted images along the timeline from randomly selected points. They also implemented masking techniques, including time masking (TM) and frequency masking (FM), where random portions of images were erased and replaced with zero values along respective axes. Additionally, brightness adjustment methods were applied, including both lower brightness (LB) and higher brightness (HB) modifications. The study found that random sampling, along with the FH+TS combination, was most effective, while FM and HB methods reduced accuracy. They extracted mel-spectrogram images of size $535 \times 678 \times 3$, resizing them afterwards to $100 \times 128 \times 3$. The study optimized the VGG-13 architecture for SI by reducing the number of convolutional layers from 10 to 5 and changing the pooling layers from max pooling to average pooling. They also added batch normalization layers after each average pooling layer and decreased the dropout layers from 10 to just 1. Furthermore, the fully connected layers were modified from three layers to a single layer with 1251 hidden neurons, matching the number of speakers in the VoxCeleb1 dataset. The study primarily utilized triple loss, n-pair loss, and angular loss, achieving a top-1 accuracy of 91.17%.

Other recent successful CNN-based approaches use different feature representations or training strategies.

Nugroho *et al.* [NN$^+$22] employed pitch shifting data augmentation to enhance Indonesian multiethnic SRcognition performance. The researchers collected a dataset of 700 audio samples from 70 ethnic male speakers, processing each with a 1-second segment at a standard sample rate of 44.1 kHz with 32-bit depth. They extracted acoustic features using Mel-Frequency Cepstral Coefficient (MFCC) and used it as input for a 7-layer Deep Neural Network architecture consisting entirely of dense layers, where the input layer contained 193 nodes matching the MFCC features, and subsequent layers progressively reduced the number of nodes by half, with dropout layers between each dense layer to prevent overfitting. The pitch shifting data augmentation approach outperformed other augmentation methods, including adding white noise and time stretching, achieving a top-1 accuracy of 99.27% on their dataset.

Fasounaki *et al.* [FYÖİ21] presented a CNN architecture specifically adapted for SI tasks, utilizing rectangular kernels to capture temporal speech characteristics. The input features were MFCCs, with each audio instance segmented into 1-second segments, resulting in input data of 13100. On the VoxCeleb1 dataset, the model attained a top-1 accuracy of 90.0%. The loss function employed was Categorical Cross Entropy, with Sigmoid activation in the middle layers and Softmax in the output layer.

Ye *et al.* [YY21] proposed a DNN model that combines a 2-D CNN with Gated Recurrent Units (GRU) for SI. The researchers utilized the Aishell-1 dataset, a Mandarin speech corpus, for their experiments. They used normal spectrograms as input for the model. For training, the model employed categorical cross-entropy as the loss function. The proposed deep GRU model achieved a top-1 accuracy of 98.96% on the Aishell-1 dataset. The study also demonstrated the model's robustness by showing that its recognition accuracy dropped less significantly than other models when Gaussian white noise was added to the testing data.

The reviewed studies share several common characteristics: most adapt or redesign convolutional architectures—frequently VGG-based, but also including ResNet, CapsuleNet, GRU

hybrids, and DNNs—to suit SI tasks. Many employ advanced or metric-learning loss functions to enhance feature discrimination and generalization. Reported top-1 accuracies span a broad range, from around 71.8% to 91.78% on standard VoxCeleb1 [NCZ17, ASSL19, ATL19, YR18] baselines to over 99% on domain-specific datasets [NN$^+$22]. VoxCeleb1 is the most widely used benchmark, though some works evaluate on alternative corpora such as Aishell-1. Across studies, a variety of data augmentation methods and time–frequency representations (e.g., different spectrogram or MFCC dimensions) are used to address short utterances, noise robustness, and class imbalance.

Building upon these established approaches, our work contributes to this research landscape by systematically studying the application of advanced margin-based loss functions (ArcFace and CosFace) compared to traditional Softmax in SI, and by maximizing their accuracy through preprocessing optimization. We implement modifications to the VGG16 architecture to implement these loss functions and experiment with multiple mel-spectrogram dimensions to create optimal conditions for loss function comparison. Additionally, we introduce an audio looping technique as a preprocessing method specifically designed to enhance temporal information capture for the VoxCeleb1 dataset.

While our approach achieved a competitive top-1 accuracy of 83.15%, we did not surpass the higher accuracies reported in the literature. This indicates that further investigation is needed to fully unlock the potential of margin-based loss functions in SI systems. This performance gap represents an important area for continued research, particularly in combining our preprocessing strategies with more augmentation techniques or architectural innovations.

CHAPTER 3

Concepts and Definitions

3.1 Concepts and Definitions

Modern artificial intelligence systems are increasingly driven by the ability to process vast amounts of data, identify underlying patterns, and learn from them to perform complex tasks. This data-driven approach allows machines to mimic cognitive functions, such as recognizing speech or, in our case, identifying speakers. The efficacy of any AI model is significantly influenced by the preprocessing of its input data and the learning algorithms it employs [SNP20]. Thus, this chapter will provide foundational concepts and definitions concerning data processing and machine learning, particularly as they relate to the challenges and techniques used in SI, setting the stage for the subsequent discussion of our proposed methodology.

3.1.1 Speech Data Processing

In the context of SI, the primary goal of speech data processing is to transform the raw audio waveform into a parametric representation that captures speaker-specific characteristics. This process, known as feature extraction, yields a format interpretable by machine learning models.

3.1.1.1 Mel-Spectrograms

For this study, we employed mel-spectrograms as the primary feature representation. Mel-spectrograms are derived from the audio signal by converting it into the frequency domain and then mapping these frequencies onto the Mel scale, which better approximates human auditory perception [Ped65].

Following feature extraction, several techniques can be applied to optimize the training process and enhance model performance, including feature selection, feature reduction, and feature scaling. While feature selection involves choosing a subset of the original features without transformation, and feature reduction involves transforming features to a lower-dimensional space, our focus was primarily on the specific pre-processing and augmentation steps that directly influenced the input to our model and the effectiveness of the different loss functions on accuracy.

3.1.2 Speaker Recognition Systems

3.1.2.1 Speaker Recognition Categories

Speaker Recognition (SR) systems can be modeled either as Text-Dependent or Text-Independent systems. Text-Dependent SR relies on the user uttering whatever is being prompted, while text-independent SR provides a more flexible approach where there is no constraint on the user of what can be said, and as such, has broader applications.

SR encompasses voice-based technologies designed to identify or verify individuals using their vocal characteristics. Although it can be divided into several subcategories, it is generally divided into three primary categories:

- **Speaker Verification (SV)**. Aims to verify whether a speaker's claimed identity is true. The system compares the speaker's voice to a stored voiceprint of the claimed identity to confirm their identity [JSR03]. SV is commonly used in authentication systems, such as secure access to privileged information, devices, or accounts [DSJM22].

- **Speaker Identification (SI)**. Aims to determine the identity of an unknown speaker from a group of known speakers. The system compares the input voice to a database of voiceprints and identifies the closest match. SI can then also be split into two approaches:

  - **Closed-set** In this scenario, the system assumes that the unknown speaker is already enrolled in the reference database. The objective is to identify which of the known speakers the input voice belongs to [BCD+22].
  - **Open-set** This approach accommodates situations where the unknown speaker may or may not be present in the reference database. The system must not only attempt to identify the speaker if they are known but also be capable of rejecting an unknown speaker if their voice does not match any enrolled profiles [Cam02]. SI finds applications in areas such as law enforcement, voice assistants, and call center analytics [Fur10].

- **Speaker Classification (SC)**. Distinguishes and classifies speakers based on specific characteristics such as age, gender, and health. It is commonly used in scenarios such as demographic analysis and targeted marketing [QMB17].

Each category faces unique challenges and applications. For instance, SV must handle variability in a person's voice due to emotional states or background noise, while SI requires robust matching algorithms to distinguish between similar voices, often necessitating large and diverse databases to improve accuracy. SC, on the other hand, requires sophisticated feature extraction techniques and machine learning models to accurately capture and analyze subtle vocal traits.

The underlying system for SI works similarly to a fingerprint matching process. Looking at the spectral content, the system can analyze the unique features of a person's voice and match it against a database. These features are highly distinctive, capturing the physiological and behavioral characteristics of an individual's voice [MOR].

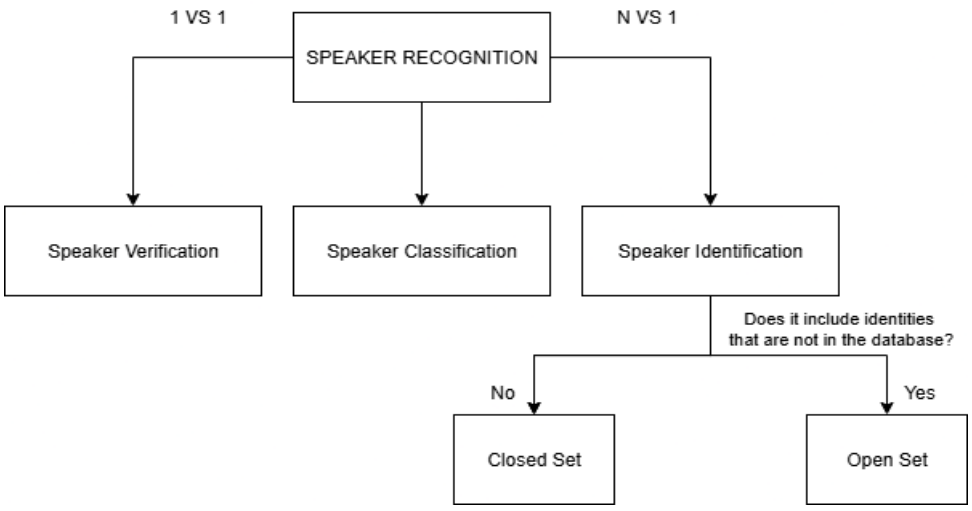

Figure 3.1 SR can be split into three main categories, SV, SC and SI. SI can be further split into Text Dependent and Text Independent, which will further split into closed set and open set. Our main focus will be on Closed Set Text Independent SI

3.1.3 Loss Functions

A loss function is a mathematical formulation that measures the discrepancy between a machine learning model's predicted output and the true target value. It quantifies the error, providing a scalar value that an optimization algorithm seeks to minimize during the training phase. By adjusting the model parameters to reduce the loss, the model learns to make more accurate predictions [TCERP+23]. For classification tasks, loss functions like Softmax, AMSoftMax, and ArcFace are commonly used. These functions evaluate the model's confidence in assigning an input to a particular class, and in the context of SI, they are designed to encourage distinct feature representations for different speakers, thereby improving identification accuracy.

3.1.4 Data Augmentation

Data augmentation is a set of techniques used to increase the size and diversity of a training dataset by creating modified copies of existing data or synthesizing new data from existing data. The primary goal of data augmentation is to improve the generalization ability of machine learning models, making them more robust to variations in input data and reducing overfitting. By artificially expanding the training set, augmentation exposes the model to a wider range of data characteristics, thereby enhancing its performance on unseen data [LLA22].

In the context of this research, we used data augmentation to address the limited audio data, particularly short utterance lengths. Ways of creating new training samples by applying transformations to existing ones incluse noise adition [KPPK15], changing pitch [AWGC23], time stretching [HKAS], or as Nugroho *et al.* [NN+22] proposed, all three of them. The technique we propose, audio looping, directly addresses the challenge of insufficient temporal information in shorter speech samples, focusing instead on capturing more speaker-specific vocal characteristics.

CHAPTER 4

Methodology

4.1 Methodology

4.1.1 Dataset

For this study, we used the VoxCeleb1 dataset [NCZ17], a widely used benchmark for SR and verification tasks. VoxCeleb1 contains over $100,000$ utterances from $1,251$ speakers, collected from publicly available interview videos on platforms such as YouTube [NCZ17].

The audio samples in VoxCeleb1 are recorded at a sample rate of 16kHz, which is standard for speech processing tasks. The dataset includes recordings in multiple languages, with speakers of diverse accents, ages, and genders, being made up of 688 males and 563 females.

The utterances are recorded in various environments, from outdoor stadiums to quiet indoor studios, with almost all of them having some form of background noise, such as laughter, background chatter, and overlapping speaking.

It is worth noting that the VoxCeleb1 dataset, while widely utilized, presents a challenge in the form of an imbalance in the number of audio samples per speaker as shown in Table 4.1. As you can see, 212 people have almost double the amount of audios than the 532 speakers with less than 90 audios per speaker. Although data augmentation techniques can be employed to mitigate such imbalances, our study specifically focused on the audio looping method for duration extension in combination with the usage of advanced loss functions. Consequently, the class imbalance remains a present factor in our experiments and represents an avenue for future research. We hypothesize that by providing the system with extended audio data per sample, it becomes more inclined to learn speaker-specific features, thereby enhancing overall system accuracy.

For our experiment, we used a train-validation-test split of 70%, 15%, 15% as illustrated in Table 4.2.

Table 4.1 Information about the database highlighting the class imbalance.

Audios Per Speaker	Number of Speakers	Number of Audios
<90	532	35431
90-180	507	63785
>180	212	54300

Table 4.2 Information about the database used for experimentation.

Parameters	Voxceleb1
Total number of speakers	1251(688 male, 563 female)
Total utterances	153,516
Train utterances	108,018
Validation utterances	23,066
Test utterances	22,432

4.1.2 Preprocessing

A lot of the audios start or end with silence. Figure 4.1 shows an example where there is a silent situation at the beginning (region with only purple color). We start by reducing this, ensuring that the system does not waste resources by processing empty data.

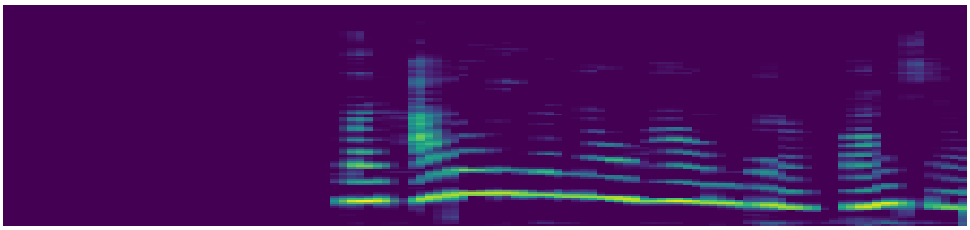

Figure 4.1 Spectrogram image of audio before silence removal.

4.1.2.1 Data Augmentation

Following the reduction of silence from the audio signals, many recordings were found to be shorter than the desired duration. To address this, we implemented an audio looping technique to extend these recordings to a required length, as illustrated in Figure 4.3. This enabled the creation of two distinct dataset versions: (a) consisting of three-second audio clips and (b) with ten-second clips. Our experimental setup utilized these target durations, with three seconds chosen as it is a common standard in SI research, and ten seconds selected based on observations of diminishing returns where extending duration further yielded no noticeable accuracy improvements.

It is important to note the specific implementation of our audio looping method: looping was applied only at the end of an audio segment if it was shorter than the target duration. If an audio sample already met or exceeded the target length (e.g., 10 seconds), no looping was applied, and for longer samples (e.g., 20 seconds), a 10-second segment was extracted. This approach means that the effectiveness of the audio looping method in more consistent environments or with different looping strategies requires further investigation.

4.1.2.2 Mel-spectrogram Generation

The process of generating mel-spectrograms involves several steps to convert the raw audio signal into a time-frequency representation that aligns with human auditory perception [BRU+19].

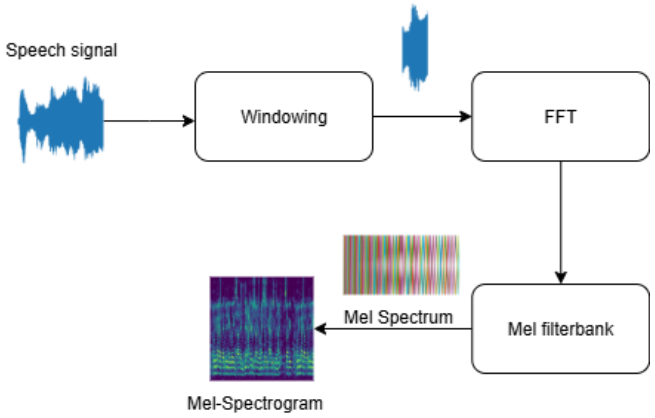

Figure 4.2 Process from start to finish showing the steps to generate a mel-spectrogram starting from a speech signal.

This is visually depicted in Figure 4.2. The detailed steps are as follows:

1. **Framing and Windowing:** The audio signal is first divided into short, overlapping segments known as frames or windows. This segmentation allows for the analysis of the signal's frequency content over time [HTL24]. While there are different windowing functions, for this study we used the Hamming window for our mel-spectrogram generation, as shown in 4.1.

2. **Fast Fourier Transform (FFT):** For each windowed segment, the Fast Fourier Transform (FFT) is applied. This operation, transforms the signal from the time domain into the frequency domain, decomposing it into its constituent frequencies and their respective amplitudes, thereby revealing the spectral content of that short time slice. The collection of these transformed segments across time constitutes the Short-Term Fourier Transform (STFT), depicted in 4.2.

3. **Mel Filter Banks:** The resulting frequency spectrum from each window is then filtered using a set of Mel filter banks. The Mel scale is a perceptual scale of pitches that approximates the non-linear frequency response of the human ear. This means it assigns more emphasis to lower frequencies and less to higher frequencies, which is crucial for realistic audio analysis and for capturing perceptually relevant features for speaker characteristics.

4. **Spectrogram Envelope:** The output from each Mel filter in the filter bank is summed and combined to create the spectrogram envelope. This envelope represents the energy or amplitude of the signal across different frequency bands on the Mel scale, evolving over time.

5. **Mel-spectrogram Visualization:** Finally, the mel-spectrogram is visualized as a 2D image. In this representation, the x-axis denotes time, the y-axis represents frequency on the Mel scale, and the color intensity at each point indicates the amplitude or energy of the signal at that specific time-frequency combination.

For a window of length N (from sample 0 to $N-1$), the Hamming window function $w(n)$ is given by 4.1 [HAR87]:

$$W(n) = 0.54 - 0.46 \times \cos \frac{(2\pi n)}{(N-1)} \quad \text{for } 0 \leq n \leq N-1 \tag{4.1}$$

Where:

- n is the sample index
- N is the window length (total number of samples)

The equation for the STFT process is defined as follows 4.2:

$$X(m,k) = \sum_{n=0}^{N-1} x(n+mH) \cdot w(n) \cdot e^{-j2\pi kn/N} \tag{4.2}$$

Where:

- $X(m,k)$ is the STFT coefficient at frame m and frequency bin k
- $x(n)$ is the input signal
- $w(n)$ is the window function (in our case, a Hamming window)
- H is the hop size
- N is the FFT size
- m is the frame index
- k is the frequency bin index

The mel-spectrogram is then obtained by applying mel filter banks to the power spectrum, as depicted in 4.3:

$$M(m,j) = \sum_{k=0}^{N/2} |X(m,k)|^2 \cdot H_j(k) \tag{4.3}$$

Where:

- $M(m,j)$ is the mel-spectrogram coefficient at frame m and mel bin j
- $|X(m,k)|^2$ is the power spectrum from the STFT
- $H_j(k)$ is the j-th mel filter bank response at frequency bin k
- $j = 0,1,2,...,n_{mels}-1$

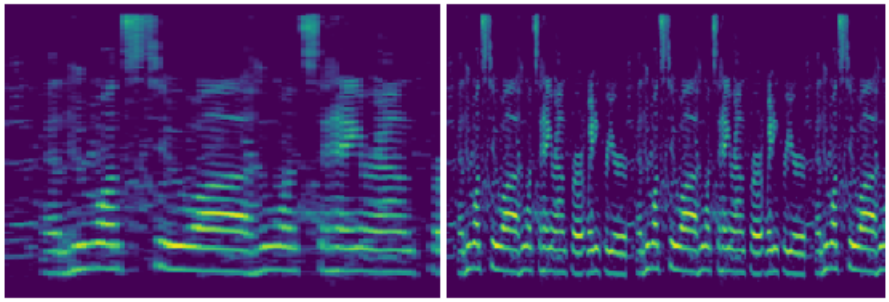

Figure 4.3 Visual representation of mel-spectrograms (a) before and (b) after looping. The looping process extends shorter audio segments to meet the required duration.

For our mel-spectrogram generation, we used a Hamming window and a hop size of $H = 512$ and FFT size of $N = 2048$ with an overlap of 75%. The resulting STFT produces frequency bins $k = 0, 1, 2, ..., 1024$, which are then processed through $n_{mels} = 128$ mel filter banks to generate the final mel-spectrogram representation. These values are fairly standard for speech processing tasks. An example of a mel-spectrogram is depicted in Figure 4.4.

Spectrograms, particularly mel-spectrograms, are robust to noise and variations in recording conditions, as shown by Nagrani *et al.* [NCZ17], Lambamo *et al.* [LSJ22], and Saritha *et al.* [SLK+24]. In these works, the authors showed that spectrograms can mitigate the effects of background noise by focusing on the frequency patterns that are most relevant to the characteristics of the speaker. The conversion of spectrograms into the Mel scale is done because the Mel scale is designed to mimic the way humans perceive sound, with a non-linear frequency resolution that assigns more weight to lower frequencies. Lower frequency components have been shown to be more distinguishing between speakers in SI systems [LD08, ZZOEA25].

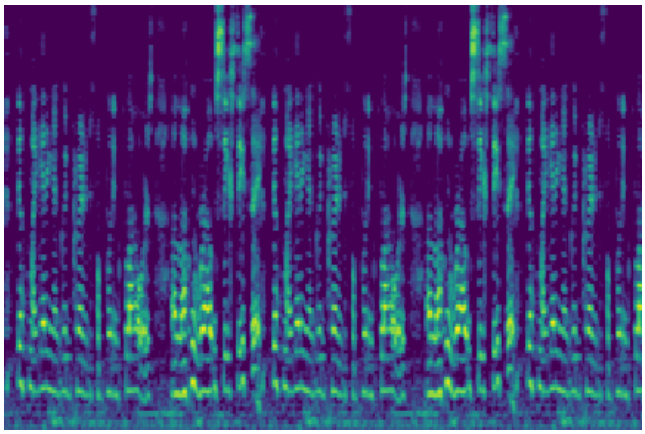

Figure 4.4 A typical Mel-spectrogram image used to represent a speech signal.

Using the audio data we previously generated, we convert the recordings into ten-second Mel-spectrograms. These are then resized into three specific dimensions: $224 \times 224 \times 3$, $448 \times$

448×3, and $432 \times 288 \times 3$. The same process is applied to the dataset of three-second audio recordings. Additionally, we apply mean and variance normalization on a per-speaker basis to ensure consistent processing across the dataset [NCZ17].

4.1.3 Loss functions

Much research has been conducted using the cross-entropy loss function, commonly known as Softmax loss [NCZ17, ATL19]. The Softmax loss function for a batch of n samples is mathematically defined as:

$$L_{\text{AMS}} = -\frac{1}{n}\sum_{i=1}^{n}\log\left(P(y=y_i|x_i;W)\right) \tag{4.4}$$

where $(P(y=y_i|x_i;W))$ is the probability of y is the class y_i, given the sample x_i and the weight vector W.

While these approaches have yielded positive results, Softmax presents several limitations for SI. A primary drawback is its failure to explicitly enforce discriminative margins between different speaker classes. This can result in learned embeddings for distinct speakers being too close in the feature space, hindering the model's ability to differentiate them, particularly under noisy conditions or with limited training data. Furthermore, Softmax does not directly optimize for intra-class compactness; while it pushes embeddings away from decision boundaries, it doesn't strongly encourage embeddings from the same speaker to cluster tightly together as seen in Figure 4.5. This can lead to more spread-out representations for a single speaker, reducing the robustness of the SI system. An additional concern is that Softmax treats all classes equally, which can be problematic in SI where the number of speakers (classes) can be very large, potentially leading to imbalanced class distributions and a bias towards more frequent speakers in the training data. This is especially true for us, as there is a large class imbalance in our dataset, as can be seen in Table 4.1.

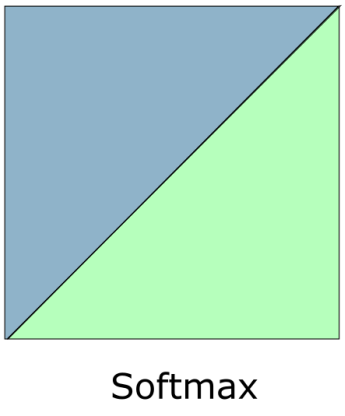

Figure 4.5 Traditional Softmax's decision boundary meets at P_0. D_1 and D_2 represent class 1 and class 2, respectively.

To address these limitations, our study investigates the effectiveness of alternative loss func-

tions: CosFace and ArcFace [WWZ+18, DGXZ19]. These functions are designed to directly improve class separation and intra-class compactness, which are crucial for robust SI. We focus on enhancing intra-class similarities by employing these functions, which, although originally developed for facial recognition, have demonstrated significant potential for SI tasks. Both CosFace and ArcFace work by enhancing intra-class similarities through the imposition of angular or margin-based constraints [WWZ+18, WCLL18, DGXZ19].

4.1.4 CosFace

CosFace Loss is an enhanced version of the traditional Softmax loss, designed to improve the discriminative power of deep neural networks by imposing a fixed angular margin between classes [WCLL18]. This is achieved by modifying the target logit through the addition of a margin to the cosine similarity score of the target class.

The CosFace loss introduces an additive margin to the cosine similarity score of the target class, formulated as:

$$\psi(x) = x - m \tag{4.5}$$

where $x = \cos\theta_{y_i}$ is the cosine similarity between the feature vector of the target class, and m is the additive margin. This formulation ensures that the decision boundary is pushed away from the target class by a fixed margin, thereby improving class separation.

The geometric interpretation of this concept is illustrated in Figure 4.6. While traditional Softmax loss places the decision boundary at a single hyperplane P_o (two-class problem), CosFace introduces a marginal region, shifting the boundary away from the target class. This shift is calculated as $m = (W_1 - W_2)^T P_1$. Where W_1 and W_2 are the weight vectors of the two classes, and P_1 is the decision boundary for class 1. This fixed angular margin leads to more discriminative and compact features.

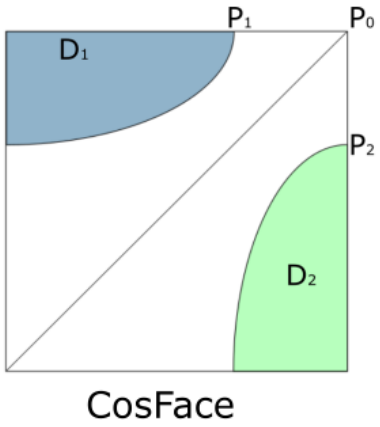

Figure 4.6 A comparison between traditional Softmax's decision boundary and CosFace's decision boundary. Softmax's decision boundary meets at P_0 whereas CosFace's decision boundary for class 1 is at P_1 and for class 2 it is at P_2. D_1 and D_2 represent class 1 and class 2, respectively.

The CosFace loss function for sample x_i is mathematically defined as:

$$L_{\text{CF}} = -\log\left(\frac{e^{s(\cos(\theta_{y_i})-m)}}{e^{s(\cos(\theta_{y_i})-m)} + \sum_{j\neq y_i} e^{s\cos\theta_j}}\right) \tag{4.6}$$

where:

- s is the scaling factor, which amplifies the separation between classes;
- m is the additive margin, which enforces a fixed angular separation;
- $\cos(\theta_{y_i})$ is the cosine similarity between the features and the target weights.

For our study, we used $s = 22$ and $m = 0.2$, which were chosen to optimize the trade-off between class separation and model convergence.

4.1.5 ArcFace

The ArcFace loss function, introduced by Deng *et al.* [DGXZ19], was originally used for face verification, but has seen some recent use in SR technology [GYG21]. This function is specifically designed to optimize the geodesic distance margin on a hypersphere by introducing an additive angular margin that pushes the decision boundary further from the target class, thereby enhancing class separation.

ArcFace optimizes speaker embeddings through a dual approach: maximizing intra-class compactness while simultaneously increasing inter-class separation as seen in Figure 4.7. This optimization strategy has seen some success in improving speaker distinction, particularly when paired with large-scale datasets where subtle differences between speakers must be accurately captured.

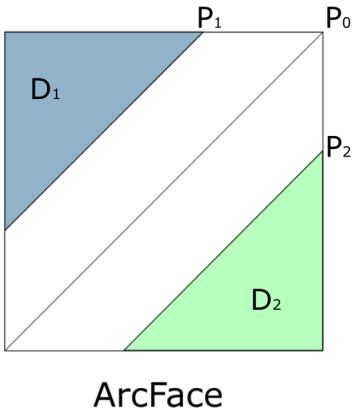

Figure 4.7 A comparison between traditional Softmax's decision boundary and ArcFace's decision boundary. Softmax's decision boundary meets at P_0 whereas ArcFace's decision boundary for class 1 is at P_1 and for class 2 it is at P_2. D_1 and D_2 represent class 1 and class 2, respectively.

The integration of ArcFace with SincNet architectures [GYG21] has enabled direct feature extraction from raw speech signals. When combined with dual attention mechanisms, this

approach delivers performance improvements, especially in short-utterance scenarios where limited audio data is available for SI.

ArcFace demonstrates the capability to effectively handle difficult operational conditions, including short utterances and cross-domain mismatches. In far-field SV applications, the loss function has been successfully adapted to address domain mismatches [LQL22], with optimized margin penalties during training significantly enhancing overall system performance.

The practical advantages of ArcFace extend to various challenging environments. Models incorporating ArcFace have shown reduced error rates in short speech scenarios, making them effective for real-world applications where audio samples are typically brief. Additionally, the enhanced feature extraction and classification capabilities contribute to superior performance in noisy environments, addressing one of the most common challenges in SI systems.

Comparative studies have demonstrated that ArcFace-based models consistently achieve lower error rates than traditional methods [THHH22], underscoring their effectiveness in SI tasks. Furthermore, the loss function's inherent ability to stabilize training processes and enhance feature learning establishes it as an effective tool in deep learning-based SR systems [THHH22].

For a sample (x_i, y_i), where x_i is the input sample and y_i is the target label, the mathematical formulation of the ArcFace loss function is as follows:

$$L_{\mathrm{AF}}(x_i) = -log\left(\frac{e^{s(\cos(\theta_i+m))}}{e^{s(\cos(\theta_i+m))}+\sum_{j\neq y_i} e^{s(\cos(\theta_j))}}\right) \tag{4.7}$$

where:

- s is the scaling factor;
- m is the additive angular margin;
- $\cos(\theta_{y_i})$ is the angle between the feature vector and the target weights.

In our implementation, the initial scaling factor s was set to $s = 22$. The additive angular margin m was set to 0.2. This configuration was chosen to ensure more compact and discriminative features.

4.1.6 Recognition Model

4.1.6.1 VGG16

The VGG16 model is a convolutional neural network (CNN) comprising 16 weight layers: 13 convolutional layers and 3 fully connected layers. These layers are organized into five blocks, each followed by a max-pooling layer to progressively reduce the spatial dimensions of the feature maps [SZ14] As can be seen in Table 4.3. The core of the architecture can be broken down as follows:

- Convolutional Layers: The first two blocks contain two convolutional layers each, while the last three blocks contain three convolutional layers each. Each convolutional layer

Table 4.3 Original VGG16 Architecture. This table details the sequential layers, kernel sizes, strides, and activation functions characteristic of the VGG16 network as described by Simonyan and Zisserman [SZ14].

Original VGG16					
Layer	Feature Map	Size	Kernel Size	Stride	Activation
Image	1	224 x 224 x3	-	-	-
2 X Convolution	64	224 x 224 x 64	3x3	1	relu
Max Pooling	64	112 x 112 x 64	3x3	2	relu
2 X Convolution	128	112 x 112 x 128	3x3	1	relu
Max Pooling	128	56 x 56 x 128	3x3	2	relu
2 X Convolution	256	56 x 56 x 256	3x3	1	relu
Max Pooling	256	28 x 28 x 256	3x3	2	relu
3 X Convolution	512	28 x 28 x 512	3x3	1	relu
Max Pooling	512	14 x 14 x 512	3x3	2	relu
3 X Convolution	512	14 x 14 x 512	3x3	1	relu
Max Pooling	512	7 x 7 x 512	3x3	2	relu
Dense	-	25088	-	-	relu
Dense	-	4096	-	-	relu
Dense	-	4096	-	-	relu
Dense	-	1000	-	-	Softmax

applies a 3×3 filter with stride 1 and padding to preserve spatial resolution, followed by a ReLU activation function;

- Pooling Layers: A max-pooling layer with a 2×2 filter and stride 2 is applied after each block to downsample the feature maps.

4.1.6.2 Modified VGG16

To enhance the model's performance and adaptability for SI, we proposed the following modifications using PyTorch, as seen in Table 4.4:

1. First, we replaced the fixed size average pooling layer with a global average pooling layer to handle inputs of varying spatial dimensions.

2. The fully connected classifier was replaced with a custom classifier designed to output embeddings of a specified dimension. The output of the convolutional layers is first flattened and passed through a fully connected layer with 1024 units, followed by a ReLU activation and dropout of 0.3 for regularization. Finally, it outputs the optimal 256-dimensional embeddings through another fully connected layer [HD18, SKP15].

3. We added a classification head. This head is a fully connected layer that maps the embeddings to the number of classes, in our case, 1251;

Table 4.4 Proposed Modified VGG16 Architecture. Tested with mel-spectrogram dimensions: 224×224×3, 448×448×3, and 432×288×3. The table highlights the proposed modifications to the VGG16 convolutional base, retaining its core feature extraction blocks. Notable adaptations include the integration of global average pooling, a custom classifier yielding 256-dimensional embeddings, and a final classification head for SI. The embeddings are further normalized using L2 normalization for cosine similarity.

Modified VGG16					
Layer	Feature Map	Size	Kernel Size	Stride	Activation
Image	1	Any Height x Any Width x 3	-	-	-
2 X Convolution	64	224 x 224 x 64	3x3	1	relu
Max Pooling	64	112 x 112 x 64	3x3	2	relu
2 X Convolution	128	112 x 112 x 128	3x3	1	relu
Max Pooling	128	56 x 56 x 128	3x3	2	relu
2 X Convolution	256	56 x 56 x 256	3x3	1	relu
Max Pooling	256	28 x 28 x 256	3x3	2	relu
3 X Convolution	512	28 x 28 x 512	3x3	1	relu
Max Pooling	512	14 x 14 x 512	3x3	2	relu
3 X Convolution	512	14 x 14 x 512	3x3	1	relu
Global Avg Pooling 2D	-	512	N/A	N/A	-
Dense	-	1024	-	-	relu
Dropout (0.3)	-	1024	-	-	-
Dense	-	256, followed by L2 normalization	-	-	relu
Dense	-	1251 (num speakers)	-	-	Softmax

4. To ensure the embeddings are normalized to unit vectors, the model is wrapped in a custom class. During the forward pass, the embeddings are normalized using L2 normalization ($||x|_2 = 1$), because we use cosine similarity to compare embeddings.

The model parameters were optimized using Stochastic Gradient Descent (SGD) with an initial learning rate of 0.001, momentum of 0.9, and weight decay of 1×10^{-4}. To adapt the learning rate during training, we employed a ReduceLROnPlateau scheduler that monitored validation loss. The scheduler reduced the learning rate by a factor of 0.1 when no improvement in validation loss was observed for 5 consecutive epochs, with a minimum improvement threshold of 1×10^{-4} to qualify as meaningful progress. We also incorporated an early stopping mechanism with a patience of 15 epochs of no improvement, while also enforcing a minimum of 30 epochs to ensure that the model had sufficient opportunity to converge, even if progress was initially slow.

CHAPTER 5

Results

5.1 Experimental results

First, we ran the CNN with the traditional SoftMax function to compare it with results of Nagrani *et al.* [NCZ17], who, as mentioned before, used random sampling as their data augmentation method with 3 second audios, and a largely unaltered VGG network, making only slight changes to accommodate their mel-spectrogram dimensions of size 512×300. As we can see in Table 5.2, Nagrani *et al.*'s solution performed better than ours when using the traditional SoftMax loss function (80.5% vs. our best Softmax result of 67.62% for 3-second clips). Given this gap, and to examine temporal context, we used the audio looping method to further augment our data by increasing our audio duration to ten seconds. While our accuracy does improve, it still underperforms compared to Nagrani *et al.*'s setup. This is in part due to:

- Our data augmentation method does not diminish class imbalance, focusing instead on temporal aspects.
- Our CNN was modified specifically to accommodate loss functions that use cosine similarity.

As such, when we use a loss function that uses cosine similarity, like ArcFace or CosFace, as seen in Table 5.1, we see a substantial improvement over Softmax. Although with the 3-second variations we still don't achieve better results compared to nagrani *et al.* (68.05% and 69.82% for ArcFace and CosFace respectively), we do see a significant jump in accuracy when we use audio looping to increase audio duration, with CosFace achieving 83.15% and surpassing nagrani *et al.*'s baseline.

5.1.1 Audio Looping Data Augmentation: Core Findings

The cornerstone of this research lies in evaluating our novel audio looping data augmentation technique across different loss functions and input configurations. Table 5.1 presents comprehensive results demonstrating the effectiveness of our temporal extension approach.

Our audio looping technique demonstrated universal effectiveness, achieving performance improvements in all 18 tested configurations (100% success rate) when extending 3-second clips to 10-second duration, as seen in Table 5.3. Notice how specifically margin-based loss functions demonstrated greater sensitivity to temporal extension compared to traditional Softmax.

Table 5.1 Results of SI accuracy for different loss functions, mel-spectrogram dimensions, and audio lengths.

Loss function	Mel-Spectrogram Image Dimensions	Audio length	Top-1 accuracy
ArcFace	$224 \times 224 \times 3$	3 seconds	66.28%
		10 seconds	81.33%
	$448 \times 448 \times 3$	3 seconds	64.15%
		10 seconds	79.32%
	$432 \times 288 \times 3$	3 seconds	68.05%
		10 seconds	80.79%
Cosface	$224 \times 224 \times 3$	3 seconds	66.51%
		10 seconds	79.42%
	$448 \times 448 \times 3$	3 seconds	64.93%
		10 seconds	79.56%
	$432 \times 288 \times 3$	3 seconds	69.82%
		10 seconds	83.15%
Softmax	$224 \times 224 \times 3$	3 seconds	65.76%
		10 seconds	75.41%
	$448 \times 448 \times 3$	3 seconds	67.62%
		10 seconds	75.37%
	$432 \times 288 \times 3$	3 seconds	67.58%
		10 seconds	76.41%

Table 5.2 Comparison between other studies who used VoxCeleb1 and ours.

Architecture type	Data Augmentation type	Input Type	Top-1 accuracy (%)
I-Vectors + SVM [NCZ17]	-	MFCC	49%
I-Vectors + PLDA + SVM [NCZ17]	-	MFCC	60.80%
I-Vectors + LogReg [CCL18]	-	MFCC	65.80%
VGG-like CNN + TAP [NCZ17]	Random Sampling 3 seconds	spectr	80.50%
VGG-11 based + Softmax loss [YR18]	Random Sampling + noise injection 3 seconds	spectr	83.50%
VGG-13 based + Softmax loss [YR18]	Random Sampling + noise injection 3 seconds	spectr	84.60%
VGG-like CNN + Self Attention + Random Sampling [ATL19]	Random Sampling 3 seconds	spectr	85.30%
ResNet-18 + Self-Attention + Random Sampling [ATL19]	Random Sampling 3 seconds	spectr	87.20%
VGG-11 based + Softmax loss + Center loss (joint) [YR18]	Random Sampling + noise injection 3 seconds	melspectr	88.30%
ResNet-34 + TAP [CCL18]	Random Sampling 3 seconds	melspectr	88.50%
ResNet-34 + SAP [CCL18]	Random Sampling 3 seconds	melspectr	89.20%
VGG-13 based + Softmax loss + Center Loss (joint) [YR18]	Random Sampling + noise injection 3 seconds	melspectr	89.50%
ResNet-34 + LDE [CCL18]	Random Sampling 3 seconds	melspectr	89.90%
VGG-16 Modified CNN + SoftMax (ours)	Audio Looping 3 seconds	melspectr	67.62%
VGG-16 Modified CNN + SoftMax (ours)	Audio Looping 10 seconds	melspectr	76.41%
VGG-16 Modified CNN + ArcFace (ours)	Audio Looping 3 seconds	melspectr	68.05%
VGG-16 Modified CNN + ArcFace (ours)	Audio Looping 10 seconds	melspectr	80.79%
VGG-16 Modified CNN + CosFace (ours)	Audio Looping 3 seconds	melspectr	69.82%
VGG-16 Modified CNN + CosFace (ours)	Audio Looping 10 seconds	melspectr	83.15%

Table 5.3 A more detailed look at the performance improvements achieved through audio looping data augmentation.

Loss Function	Dimensions	3-sec Acc	10-sec Acc	Improvement	Avg Improvement:
ArcFace	224×224×3	66.28%	**81.33%**	+15.05%	+14.32%
	432×288×3	**68.05%**	80.79%	+12.74%	
	448×448×3	64.15%	79.32%	+15.17%	
CosFace	224×224×3	66.51%	79.42%	+12.91%	+13.62%
	432×288×3	**69.82%**	**83.15%**	+13.33%	
	448×448×3	64.93%	79.56%	+14.63%	
Softmax	224×224×3	65.76%	75.41%	+9.65%	+8.74%
	432×288×3	67.58%	**76.41%**	+8.83%	
	448×448×3	**67.62%**	75.37%	+7.75%	

5.1.2 Performance Across Configurations

Examining Table 5.3 also reveals several consistent patterns across loss functions:

- The 432×288×3 dimensions consistently achieves the highest or near-highest performance for each loss function, suggesting this aspect ratio captures optimal spectral-temporal information.

- CosFace demonstrates the most consistent performance across different dimensions, with less variation compared to ArcFace.

- All loss functions show diminished returns with the 448×448×3 configuration, indicating that simply increasing resolution does not guarantee improved performance, and in the case of ArcFace even hinders it.

5.1.3 Comparison with Literature Strategies

5.1.3.1 Input types

The comparative analysis presented in Table 5.2 reveals the superiority of mel-spectrogram representations over traditional MFCC-based approaches in our experimental context. While MFCC-based methods in the literature achieved accuracies ranging from 49% to 65.8%, our mel-spectrogram approach achieved a superior performance even without audio looping. This performance difference aligns with findings suggesting that mel-spectrograms preserve more comprehensive spectral-temporal information compared to the compressed feature representation of MFCCs [CCL18, YR18].

It's important to note that the MFCC approaches in the literature utilized different architectures (I-Vectors with SVM/PLDA) rather than deep learning methods, making direct comparison challenging. Nevertheless, the performance gap suggests that the combination of mel-spectrogram representations with CNN architectures provides a more effective framework for capturing the complex acoustic patterns necessary for robust speaker identification.

While spectrogram-based setups in the literature achieved better results than ours (87.2% vs. our 83.15%) [YR18, ATL19], it is important to note that other setups achieved superior performance through the usage of mel-spectrograms, with accuracies ranging from 88.3% to 89.9% [CCL18, YR18]. This pattern suggests that mel-spectrogram representations offer inherent advantages over standard spectrograms for speaker identification tasks.

Our failure to achieve the highest performance levels seen in mel-spectrogram literature stems from our choice to not focus on optimizing, as such we face dataset imbalance issues, as our audio looping technique does not address the significant class imbalance present in VoxCeleb1, and our architectural choices, which prioritized compatibility with margin-based loss functions over performance optimization.

The consistent superiority of mel-spectrogram approaches across different studies reinforces the value of perceptually-motivated feature representations, even when our implementation may not achieve state-of-the-art results due to other limiting factors.

5.1.3.2 Data Augmentation

The comparative analysis also reveals the different approaches to data augmentation within the VoxCeleb1 literature, with important implications for understanding the relative contributions of different system components. The literature demonstrates a clear dominance of random sampling strategies [ATL19, YR18, NCZ17], which provide implicit class balancing benefits by increasing sampling diversity across speakers, particularly effective when sufficient audio duration per speaker is available. Our audio looping technique represents a fundamentally different augmentation philosophy, specifically designed to address temporal constraints rather than class distribution issues.

Our best-performing configuration (CosFace + 10-second looping: 83.15%) achieves several notable benchmarks within the existing literature. Most significantly, it surpasses the foundational VGG baseline established by Nagrani *et al.* (80.5%) [NCZ17] while closely matching the performance of more complex approaches such as the VGG-11 + noise injection method (83.5%) employed by Yadav and Rai [YR18]. This near-equivalent performance is particularly noteworthy given the differences in augmentation complexity.

The performance achieved through our simplified approach highlights an important practical consideration: audio looping provides comparable augmentation benefits to established techniques while requiring little computational overhead. Unlike methods that demand additional noise samples, our approach achieves substantial performance improvements through straightforward temporal repetition. This simplicity makes our method highly accessible for implementation across diverse computational environments and resource-constrained scenarios, potentially broadening the practical applicability of effective SI systems.

As expected, our approach falls short of other methods in the literature, which achieve accuracies ranging from 84.6% to 89.9%. This performance gap is partially attributable to the class imbalance preservation inherent in our audio looping strategy. While our method effectively addresses temporal constraints, it does not provide the implicit class balancing benefits that random sampling strategies offer, particularly critical in datasets like VoxCeleb1 where speaker representation and audio length varies dramatically.
The consistent superiority of random sampling approaches in high-performing literature suggests that class balance considerations may represent a more significant performance factor than temporal augmentation alone.

5.1.3.3 Architecture Findings

Another finding that emerged from examining the relationship between network architecture depth and performance outcomes. Our initial hypothesis that deeper networks (VGG16) would yield superior performance appears to conflict with the empirical evidence presented in Table 5.2. Several approaches employing less deep architectures (VGG-11, VGG-13) achieve superior performance compared to our VGG16-based implementation, suggesting that optimal network depth for SI tasks may differ from conventional wisdom established in general computer vision applications.

However, the literature results also highlight the effectiveness of ResNet architectures, with ResNet-34 achieving some of the highest accuracies [CCL18]. This suggests that the issue may not be depth per se, but rather how information flows through deep networks. ResNet's residual connections enable deeper architectures while preserving fine-grained spectral details crucial for speaker identification, which traditional deep CNNs like VGG may lose through progressive feature transformation. The success of ResNet indicates that architectural design—specifically how information is preserved and combined—may be more critical than absolute depth for speaker identification tasks.

CHAPTER 6

Conclusion

6.1 Conclusion

6.1.1 Achievement of Research Objectives

Our study comprehensively achieved all stated objectives.

- We successfully evaluated the comparative effectiveness of the ArcFace and CosFace loss functions against traditional Softmax loss and identified their superior performance when used with our modified VGG16 architecture.
- We developed and validated an audio looping technique that effectively extends shorter audio samples, proving that temporal information enhancement can overcome duration limitations in practical applications.
- We systematically identified optimal mel-spectrogram dimensions (432×288×3) that balance computational efficiency with identification accuracy.

6.1.2 Practical Implications and Applications

The enhanced accuracy and robustness achieved through our approach could benefit security systems requiring reliable voice authentication, improve virtual assistant personalization capabilities, and support forensic voice analysis applications. The audio looping technique particularly addresses practical challenges where only short utterances are available, making the system more applicable to real-world scenarios where extended speech samples are not always obtainable.

6.1.3 Study Limitations and Considerations

It is important to contextualize objectives and findings of the present study within its intended research scope. This investigation was not designed as an attempt to achieve state-of-the-art performance on the VoxCeleb1 dataset, but rather as a systematic exploration of audio looping behavior across different system configurations. Our primary research objective centered on understanding how temporal augmentation through audio looping interacts with various loss functions and input dimensions, providing foundational insights for future optimization efforts.

The experimental design prioritized comprehensive configuration space exploration over performance maximization. By systematically evaluating audio looping across three distinct loss functions (Softmax, ArcFace, CosFace), multiple mel-spectrogram dimensions (224 $\times$

224×3, $448 \times 448 \times 3$, $432 \times 288 \times 3$), and two temporal durations (3-second, 10-second), we established a controlled framework for understanding the behavioral characteristics of our proposed augmentation method. This systematic approach enables reliable conclusions about the relative effectiveness of different configurations and provides guidance for practitioners considering similar approaches.

The choice to maintain consistent architectural and preprocessing approaches across all experiments, while potentially limiting absolute performance, ensures that observed differences can be attributed to the specific variables under investigation rather than confounding factors.

Our comparison with existing literature serves as a contextual framework for understanding where audio looping fits within the broader spectrum of augmentation strategies. The competitive performance achieved relative to some established methods (notably matching Yadav and Rai's 83.5% within 0.35% [YR18]) demonstrates that audio looping represents a viable augmentation approach worthy of further development.

The identification of optimal input dimensions ($432 \times 288 \times 3$) and the superior performance of CosFace over ArcFace in this context represent practical contributions that can inform future system design decisions, even as absolute performance optimization remains a goal for subsequent research phases.

6.1.4 Future Research Directions

The audio looping technique warrants systematic investigation to establish optimal operational parameters. Questions include determining the minimum viable audio duration for effective looping and the maximum extension length before performance plateaus. Future research should explore alternative looping strategies beyond simple repetition, such as partial overlap looping or intelligent looping that prioritizes high-energy audio segments. Additionally, adaptive algorithms that adjust looping patterns based on spectral content analysis could yield superior results compared to full-segment repetition.

Combining audio looping with established techniques could address multiple system limitations simultaneously. Research should investigate optimal combinations of audio looping with random sampling for class balancing, noise injection for robustness, and other modifications. Adaptive systems that select augmentation strategies based on per-speaker data availability represent a particularly promising direction for addressing both class imbalance and temporal constraints within unified frameworks.

The findings regarding network depth suggest systematic architectural investigation could yield improvements. Future research should explore modern architectures—including transformers and efficient convolutional designs—specifically combined with audio looping. Additionally, developing networks explicitly designed to leverage the temporal patterns created by looping could achieve superior performance compared to standard architectures applied to extended inputs.

Systematic evaluation across datasets featuring different languages, acoustic conditions, and speaker demographics would establish the method's robustness and identify potential limitations. Cross-linguistic studies examining effectiveness across different phonetic structures and prosodic patterns are particularly important for understanding universal applicability.

Extension to open-set SI scenarios would broaden practical applicability, though this re-

quires investigating how temporal extension affects unknown speaker rejection capabilities. The development of threshold optimization strategies and uncertainty quantification techniques specifically designed for looped audio inputs could address deployment challenges while maintaining identification benefits.

6.1.5 Concluding Remarks

In conclusion, this research provides a systematic evaluation of margin-based loss functions in conjunction with an audio looping data augmentation strategy, within the framework of CNN-based SI. Achieving a top-1 accuracy of 83.15% on the VoxCeleb1 dataset, our system delivers competitive performance within the existing literature range. The primary value of this work lies in the proposed audio looping technique which, despite its simplicity, offers a practical and easily implementable solution for extending limited audio samples. This method is particularly well-suited for resource-constrained applications, and its integration with advanced loss functions demonstrates a viable pathway for improving SI accuracy in scenarios with short-duration recordings.

6.2 Special Thanks

I would like to formally express my gratitude to Dr. João Batista Crispiniano Garcia for his generous research grant that made this research possible.

Bibliography

[ASSL19] Prashant Anand, Ajeet Kumar Singh, Siddharth Srivastava, and Brejesh Lall. Few shot speaker recognition using deep neural networks, 2019.

[ATL19] Nguyen Nang An, Nguyen Quang Thanh, and Yanbing Liu. Deep cnns with self-attention for speaker identification. *IEEE access*, 7:85327–85337, 2019.

[AWB+21] Hadi Abdullah, Kevin Warren, Vincent Bindschaedler, Nicolas Papernot, and Patrick Traynor. Sok: The faults in our asrs: An overview of attacks against automatic speech recognition and speaker identification systems. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 730–747, 2021.

[AWGC23] Ashish Alex, Lin Wang, Paolo Gastaldo, and Andrea Cavallaro. Data augmentation for speech separation. *Speech Communication*, 152:102949, 2023.

[BCD+22] Bidhan Barai, Tapas Chakraborty, Nibaran Das, Subhadip Basu, and Mita Nasipuri. Closed-set speaker identification using vq and gmm based models. *International Journal of Speech Technology*, 25(1):173–196, 2022.

[Bot10] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pages 177–186. Springer, 2010.

[BRU+19] Abdul Malik Badshah, Nasir Rahim, Noor Ullah, Jamil Ahmad, Khan Muhammad, Mi Young Lee, Soonil Kwon, and Sung Wook Baik. Deep features-based speech emotion recognition for smart affective services. *Multimedia Tools and Applications*, 78:5571–5589, 2019.

[Cam02] Joseph P Campbell. Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 2002.

[CCL18] Weicheng Cai, Jinkun Chen, and Ming Li. Exploring the encoding layer and loss function in end-to-end speaker and language recognition system. *arXiv preprint arXiv:1804.05160*, 2018.

[DGXZ19] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[DSJM22] Mohit Dua, Ambika Sadhu, Anisha Jindal, and Raman Mehta. A hybrid noise robust model for multireplay attack detection in automatic speaker verification systems. *Biomedical Signal Processing and Control*, 74:103517, 2022.

[Fur10] Sadaoki Furui. Speaker recognition in smart environments. In *Human-centric interfaces for ambient intelligence*, pages 163–184. Elsevier, 2010.

[FYÖİ21] Mandana Fasounaki, Emirhan Burak Yüce, Serkan Öncül, and Gökhan İnce. Cnn-based text-independent automatic speaker identification using short utterances. In *2021 6th international conference on computer science and engineering (UBMK)*, pages 413–418. IEEE, 2021.

[GYG21] Miao Guo, Jiaxiong Yang, and Shu Gao. Speaker recognition method for short utterance. In *Journal of physics: conference series*, volume 1827, page 012158. IOP Publishing, 2021.

[HAR87] FREDERIC J. HARRIS. Chapter 3 - multirate fir filters for interpolating and desampling. In Douglas F. Elliott, editor, *Handbook of Digital Signal Processing*, pages 173–287. Academic Press, San Diego, 1987.

[HD18] Mahdi Hajibabaei and Dengxin Dai. Unified hypersphere embedding for speaker recognition. *arXiv preprint arXiv:1807.08312*, 2018.

[HKAS] Nina Hosseini-Kivanani, Homa Asadi, and Christoph Schommer. Speaker verification enhancement via speaking rate dynamics in persian speechprints.

[HTL24] Guoqiang Hu, Huaning Tan, and Ruilai Li. A mel spectrogram enhancement paradigm based on cwt in speech synthesis. In *2024 International Conference on Asian Language Processing (IALP)*, pages 401–405. IEEE, 2024.

[JSR03] Biing Hwang Juang, M Mohan Sondhi, and Lawrence R Rabiner. Digital speech processing. 2003.

[KKIH10] SM Kamruzzaman, ANM Karim, Md Saiful Islam, and Md Emdadul Haque. Speaker identification using mfcc-domain support vector machine. *arXiv preprint arXiv:1009.4972*, 2010.

[KPPK15] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio augmentation for speech recognition. In *Interspeech*, volume 2015, page 3586, 2015.

[LD08] Xugang Lu and Jianwu Dang. An investigation of dependencies between frequency components and speaker characteristics for text-independent speaker identification. *Speech communication*, 50(4):312–322, 2008.

[LLA22] Khaled Lounnas, Mohamed Lichouri, and Mourad Abbas. Analysis of the effect of audio data augmentation techniques on phone digit recognition for algerian

arabic dialect. In *2022 International Conference on Advanced Aspects of Software Engineering (ICAASE)*, pages 1–5. IEEE, 2022.

[LQL22] Yuke Lin, Xiaoyi Qin, and Ming Li. Cross-domain arcface: Learnging robust speaker representation under the far-field speaker verification. In *Proc. FFSVC 2022*, pages 6–9, 2022.

[LSJ22] Wondimu Lambamo, Ramasamy Srinivasagan, and Worku Jifara. Analyzing noise robustness of cochleogram and mel spectrogram features in deep learning based speaker recognition. *applied sciences*, 13(1):569, 2022.

[M+21] André Filipe da Silva Magalhães et al. Voice recognition of users for virtual assistant in industrial environments. Master's thesis, 2021.

[MOR] DAMIAN A MORANDI. Effect of pitch modification on the voice identification of the speakers.

[NCZ17] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017.

[NN+22] Kristiawan Nugroho, Edi Noersasongko, et al. Enhanced indonesian ethnic speaker recognition using data augmentation deep neural network. *Journal of King Saud University-Computer and Information Sciences*, 34(7):4375–4384, 2022.

[Ped65] Paul Pedersen. The mel scale. *Journal of Music Theory*, 9(2):295–308, 1965.

[QMB17] Zakariya Qawaqneh, Arafat Abu Mallouh, and Buket D Barkana. Deep neural network framework and transformed mfccs for speaker's age and gender classification. *Knowledge-Based Systems*, 115:5–14, 2017.

[SKP15] 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*, pages 815–823, 2015.

[SLK+24] Banala Saritha, Mohammad Azharuddin Laskar, Anish Monsley Kirupakaran, Rabul Hussain Laskar, Madhuchhanda Choudhury, and Nirupam Shome. Deep learning-based end-to-end speaker identification using time–frequency representation of speech signal. *Circuits, Systems, and Signal Processing*, 43(3):1839–1861, 2024.

[SNP20] V Sindhu, S Nivedha, and M Prakash. An empirical science research on bioinformatics in machine learning. *J. Mech. Continua Math. Sci*, 2:86–94, 2020.

[SNRM25] M Sharif-Noughabi, S Razavi, and S Mohamadzadeh. Improving the performance of speaker recognition system using optimized vgg convolutional neural network and data augmentation. *International Journal of Engineering*, 38(10):2414–2425, 2025.

[SSS+10] MS Sinith, Anoop Salim, K Gowri Sankar, KV Sandeep Narayanan, and Vishnu Soman. A novel method for text-independent speaker identification using mfcc and gmm. In *2010 International Conference on Audio, Language and Image Processing*, pages 292–296. IEEE, 2010.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[TCERP+23] Juan Terven, Diana M Cordova-Esparza, Alfonso Ramirez-Pedraza, Edgar A Chavez-Urbiola, and Julio A Romero-Gonzalez. Loss functions and metrics in deep learning. *arXiv preprint arXiv:2307.02694*, 2023.

[THHH22] Yuwu Tang, Ying Hu, Liang He, and Hao Huang. A bimodal network based on audio–text-interactional-attention with arcface loss for speech emotion recognition. *Speech Communication*, 143:21–32, 2022.

[WCLL18] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[WWZ+18] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.

[YR18] Sarthak Yadav and Atul Rai. Learning discriminative features for speaker identification and verification. In *Interspeech*, pages 2237–2241, 2018.

[YY21] Feng Ye and Jun Yang. A deep neural network model for speaker identification. *Applied Sciences*, 11(8):3603, 2021.

[ZZOEA25] Youssef Zouhir, Mohamed Zarka, Kaïs Ouni, and Lilia El Amraoui. Power wavelet cepstral coefficients (pwcc): An accurate auditory model-based feature extraction method for robust speaker recognition. *IEEE Access*, 2025.