AUTOMATED SPEAKER RECOGNITION IN NON-IDEAL AUDIO SIGNALS USING DEEP NEURAL NETWORKS

By

Anurag Chowdhury

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science – Doctor of Philosophy

2021

ABSTRACT

AUTOMATED SPEAKER RECOGNITION IN NON-IDEAL AUDIO SIGNALS USING DEEP NEURAL NETWORKS

By

Anurag Chowdhury

Speaker recognition entails the use of the human voice as a biometric modality for recognizing individuals. While speaker recognition systems are gaining popularity in consumer applications, most of these systems are negatively affected by non-ideal audio conditions, such as audio degradations, multi-lingual speech, and varying duration audio. This thesis focuses on developing speaker recognition systems robust to non-ideal audio conditions.

Firstly, a 1-Dimensional Convolutional Neural Network (1D-CNN) is developed to extract noise-robust speaker-dependent speech characteristics from the Mel Frequency Cepstral Coefficients (MFCC). Secondly, the 1D-CNN-based approach is extended to develop a triplet-learning-based feature-fusion framework, called 1D-Triplet-CNN, for improving speaker recognition performance by judiciously combining MFCC and Linear Predictive Coding (LPC) features. Our hypothesis rests on the observation that MFCC and LPC capture two distinct aspects of speech: speech perception and speech production. Thirdly, a time-domain filterbank called DeepVOX is learned from vast amounts of raw speech audio to replace commonly-used hand-crafted filterbanks, such as the Mel filterbank, in speech feature extractors. Finally, a vocal style encoding network called DeepTalk is developed to learn speaker-dependent behavioral voice characteristics to improve speaker recognition performance. The primary contribution of the thesis is the development of deep learning-based techniques to extract discriminative, noise-robust physical and behavioral voice characteristics from non-ideal speech audio. A large number of experiments conducted on the TIMIT, NTIMIT, SITW, NIST SRE (2008, 2010, and 2018), Fisher, VOXCeleb, and JukeBox datasets convey the efficacy of the proposed techniques and their importance in improving speaker recognition performance in non-ideal audio conditions.

Dedicated to Maa, Baba, and Pluto

ACKNOWLEDGMENTS

As I approach the end of my Ph.D. journey of almost five years, I would like to acknowledge the support and guidance of few very important people who were instrumental in shepherding this journey. First, I would like to thank my advisor, Dr. Arun Ross, for providing me with this wonderful opportunity to pursue my Ph.D. under his tutelage. His ability to attentively listen, logically analyze, and lucidly explain his opinions on my research problems not only improved my understanding of my research but also established a role model for a research temperament that I pursue onward. Next, I would like to thank Dr. Xiaoming Liu, Dr. Vishu Boddeti, Dr. Joyce Chai, and Dr. Fathi Salem for serving on my Ph.D. committee. Their thoughtful feedback and constructive suggestions throughout my Ph.D. have helped me develop a broader understanding of my research. They also helped me understand the real-world challenges faced in my area of research and thus guided me to develop methodologies to mitigate them. I would also like to thank Dr. Mayank Vatsa and Dr. Richa Singh for introducing me to a career in research and supporting me to pursue my Ph.D., which I now look to complete.

I have been fortunate to receive support and guidance from many exceptional researchers during my Ph.D. However, some of them have consistently helped me with their professional wisdom and friendly counsel. For that, I would like to thank all my labmates at the iPRoBe lab. Specifically, I would like to thank Jamal, Eric, Denny, Vahid, Darshika, Steven, Sudipta, Thomas, Melissa, Renu, Shivangi, and Austin for their professional advice and the camaraderie we share.

Two people who have the biggest contribution in enabling me to pursue my professional goals and have always supported me unconditionally in all my endeavors are my parents. Their love, care, support, and guidance have always motivated me to dream big and supplemented it with the courage to pursue them. Thank you, Ma and Baba. I have also been very fortunate to have found myself in the company of the best possible group of friends here in East Lansing. It is truly because of them that I never felt too far away from home. Specifically, I would like to thank Abhijnan, Atri, Reja, and Priyankar for always being around and I will miss hanging out with them. While I cannot

list all my friends here, I do owe my gratitude to them. However, of all my friends, one of them had the most valuable contribution towards my growth as a researcher and a person. For that, I would like to thank Dipti, my better half, for always being by my side throughout the process. While on the one hand, her expert professional advice helped me better understand the nuances between good and extraordinary research; on the other hand, her thoughtful advice helped me navigate several life challenges steadfastly.

I would also like to thank the CSE department for hosting and nurturing me as a Ph.D. student in the department and supporting me with any help I needed over the years. Specifically, I would like to thank Brenda Hodge, Steve Smith, Amy King, and Erin Dunlop for their help with all the administrative affairs. Finally, I owe my gratitude to MSU for providing and hosting the state-of-the-art resources essential for my research and providing a warm and inclusive Spartan community that always made me feel at home.

Go Green! Go White!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Portions of this chapter appeared in the following publication:
Chowdhury, Anurag, and Arun Ross. “Voice Biometrics and its Role in Multi-biometric Systems" ACM Computing Surveys (CSUR) (2021- To be submitted).

1.1 Biometrics

Biometrics is the science of using physical and behavioral traits to recognize humans. As a field, biometrics has been formally studied for a little less than a century [79]. However, humans have been recognizing each other using physical and behavioral traits for many millennia, making them both a subject and an example of a biometric recognition system.

In the past decade, biometric technology has become ubiquitous in the form of automatic biometric recognition systems, such as automatic fingerprint recognition, for securing smartphones [7]. A typical automatic biometric recognition system works by comparing physical or behavioral traits, also known as biometric modalities, of an individual against a gallery of enrolled users. The comparison ascertains their identity (biometric identification) or matches their biometric modality against a claimed identity (biometric verification). Every biometric recognition system has the following four stages of operation:

- **Data Acquisition:** The first step to biometric recognition is acquiring the biometric data from the chosen modality. For example, a face recognition system first acquires facial imagery from the subject of interest.

- **Feature Extraction:** The acquired biometric modality is then processed to extract a set of discriminative subject-dependent characteristics (or features). For example, in a speaker recognition system, this could be the information about the fundamental frequency and the formants of the speaker’s voice,

- **Matching:** Once extracted, the features are then compared against feature templates, from either multiple gallery subjects or a single claimed identity, to provide corresponding match score(s).

- **Decision:** The best performing match is then declared as the final result of the system, yielding an identity or verification of an identity claim as an output.

The choice of the physical or behavioral trait, i.e. the biometric modality, is vital for representing and recognizing an individual. The chosen trait can be used as a biometric modality if and only if it satisfies the following requirements:

1. Universality: Every person should possess it.

2. Uniqueness: It should be distinguishable across different people.

3. Permanence: It should not change significantly due to varying environmental factors and the passage of time.

4. Collectability: It should be measurable quantitatively.

5. Performance: It should provide practically usable recognition accuracy and speed.

6. Acceptability: Its collection and use for establishing identity should be acceptable to the general population.

7. Circumvention: It should not be easy to spoof.

The choice of biometric modality, further, depends on the application scenario of biometric recognition. For example, in a telephone banking application, the human voice is the only suitable modality for performing biometric recognition. In contrast, for an unconstrained surveillance scenario, such as in border control, face and gait are the suitable modalities, as they do not require active participation by the subjects under surveillance. The application of biometrics has also expanded to consumer applications. Consumer devices, such as smartphones and laptops, now come

Figure 1.1: Different applications of speaker recognition

equipped with biometric recognition systems. For instance, fingerprint in Touch ID, face in Face ID, and both face and fingerprint in Windows Hello are used to secure access to the device [7, 18].

1.2 Speaker Recognition: Voice as a Biometric Modality

Human voice, like fingerprint and iris patterns, is assumed to be unique across individuals. The biometric utility of human voice and its applications are studied by the field of speaker recognition, also known as voice or talker recognition.

Speaker recognition, in general, can be done by (1) listening to the speech audio (by humans), (2) visual comparison of spectrograms (by humans), or (3) automated machine-driven techniques [71], also known as automatic speaker recognition (Figure 1.2). In a typical automatic speaker recognition system, the subject speaks a phrase into a microphone to be either identified as one of the enrolled users or be verified against a claimed identity. In the case of a text-dependent speaker recognition system, a fixed phrase is provided by the system. In a text-independent speaker recognition system, no such fixed phrases are necessary, and the system recognizes the user from their vocal acoustics.

The advent of smart-devices introduced a wide variety of applications (Figure 1.1) of speaker recognition, ranging from e-commerce and personalized user interfaces to surveillance and foren-

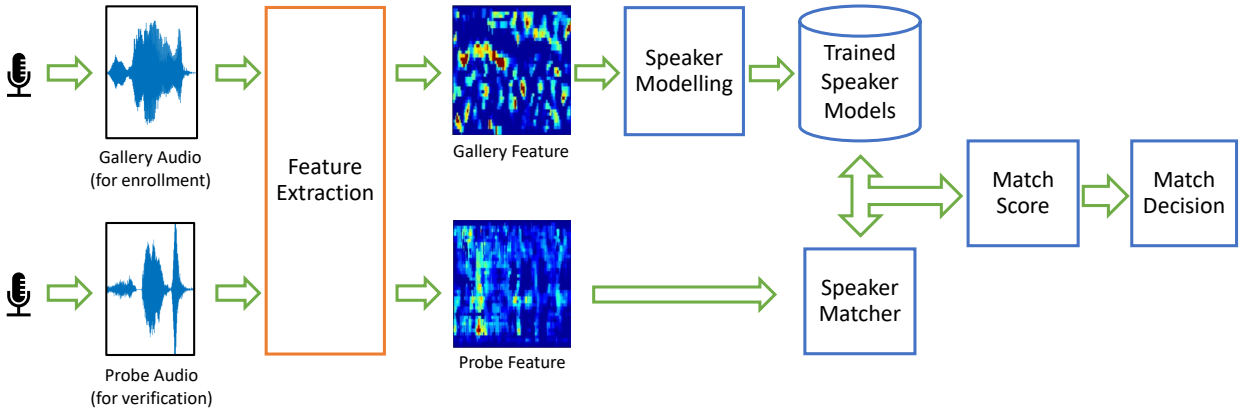


Figure 1.2: An illustration of an automatic speaker recognition system.

sics. One of the key applications of speaker recognition is securing devices with voice-controlled user interfaces (VUIs), such as digital voice assistants [14] and telephone banking systems [8, 10, 17]. VUIs are gaining popularity due to the ease-of-access provided by their hands-free operation. Such interfaces are being steadily adopted in consumer devices for improving accessibility for users with physical disabilities [6], thus widening the scope of utility of speaker recognition.

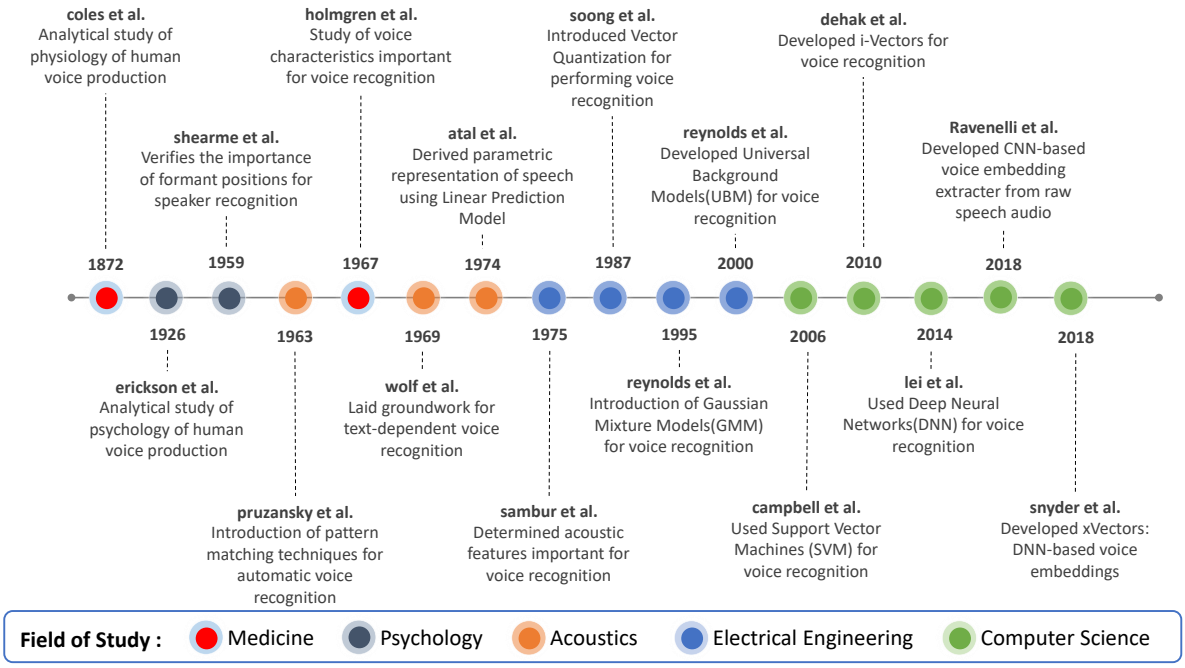


Figure 1.3: A timeline of key developments in the field of voice biometrics.

Table 1.1: A tabular representation of existing speech feature representations used for speaker recognition, as categorized by Kinnuen et al. [91].

Paper	Feature Category	Feature Details	Comments
Davis and Mermelstein [46]	Short-term spectral feature	Mel-frequency Cepstral Coefficients (MFCC)	Useful for modeling vocal tract shape
Zhao et al. [176]		Gammatone Frequency Cepstral Coefficients (GFCC)	
Mammone et al. [101]		Linear Predictive Coding (LPC)	
Huang et al. [77]		Linear Predictive Cepstral Coefficients (LPCCs)	
Hermansky et al. [72]		Perceptual Linear Prediction (PLP) coefficients	
Huang et al. [77]		Line Spectral Frequencies (LSF)	
Mitra et al. [115]		Medium Duration Modulation Cepstral (MDMC) features	
Kim et al. [87]		Power-Normalized Cepstral Coefficient (PNCC)	
Sadjadi et al. [141]		Mean Hilbert Envelope Coefficient (MHEC)	
Zheng et al. [178]	Vocal source features	Wavelet Octave Coefficients of Residues (WOCOR)	Useful for characterizing the glottal excitation signal
Gudnason et al. [65]		Voice Source Cepstrum Coefficients (VSCC)	
Kinnunen [89]	Prosodic features	Logarithmic Fundamental Frequency (F0) features	Useful for modeling speaking style of a speaker
Ferrer et al. [59]		Joint Factor Analysis based Prosody Modeling	
Doddington [52]	High Level Features	Idiolectal features	Useful for modeling lexicon of a speaker

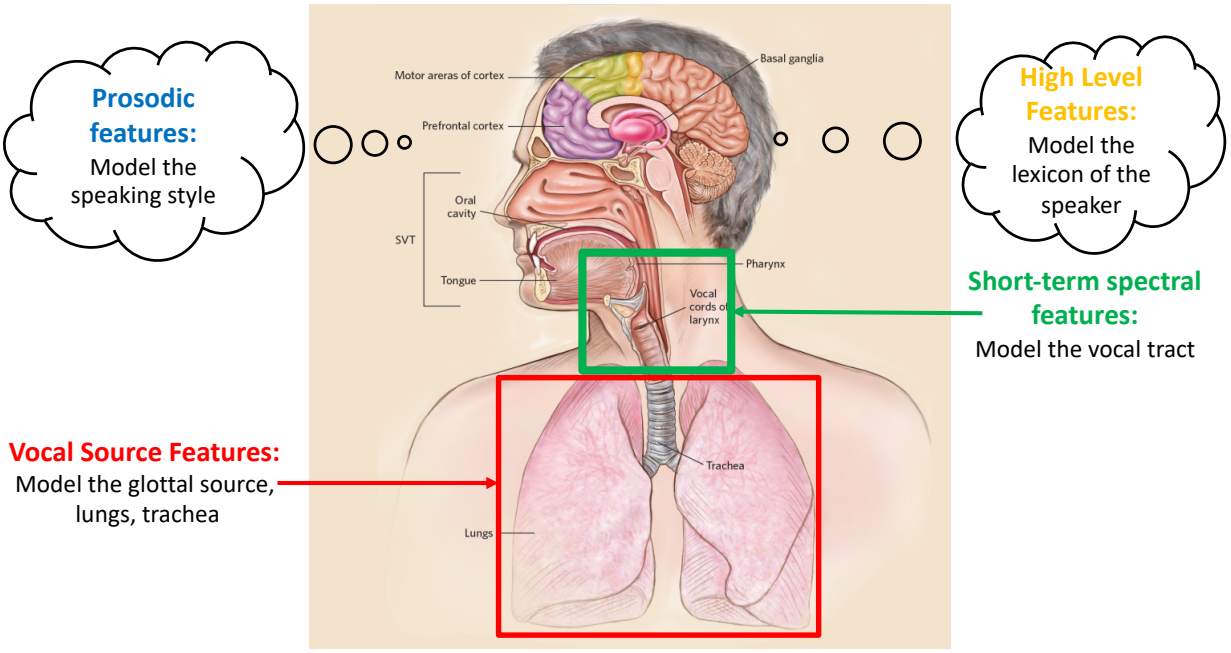


Image sourced from: https://www.the-scientist.com/features/why-human-speech-is-special--64351

Figure 1.4: Different types of voice features used for characterising a speaker.

1.2.1 Speaker dependent speech characteristics

The uniqueness of human voice, unlike fingerprint and iris modalities, is a combination of both the physical and behavioral traits of an individual. The physical characteristics of the human voice are mostly given by the size and shape of the vocal tract. On the other hand, the behavioral characteristics are encoded in the speaking style of the speaker.

The production and perception of the human voice and its use as a biometric modality were first studied in the early 1900s in the fields of human psychology and medicine (Figure 1.3) [54, 55].

The field of acoustics further studied the different factors of variability in the human voice, such as pitch, intensity, and timbre, for characterizing the human voice [23, 131, 170]. However, the scope and effectiveness of different speech processing applications were limited by the capability of analog signal processing techniques then.

The era of electronics ushered in an arsenal of digital signal processing techniques and bootstrapped the field of digital speech processing. One of the first applications of digital speech processing was to identify words and phrases in spoken language, popularly known as speech recognition. This led to the development of a variety of speech feature representations useful for performing speech recognition; the widely popular Mel-frequency cepstral coefficients (MFCC) is one such example. Although the MFCC was initially proposed to perform monosyllabic word recognition [46], it was later found to be efficient for performing speaker recognition as well [134]. However, its sensitivity to audio degradations reduced its effectiveness in speaker recognition tasks [66]. This challenge motivated the development of speech features specialized for performing speaker recognition in noisy speech audio, as summarized in Table 1.1.

1.2.2 Types of speech characteristics

In the past few decades, specialized speech features have been developed for encoding different physical and behavioral properties of the human voice. Based upon the type of voice characteristics they encode, these features have been categorized as follows [91] (Figure 1.4):

- Vocal Source Features [91]: are used to characterize the source of the human voice, which originates in the form of glottal excitation pulses.
- Short-term spectral features [113]: are used to encode the shape of the human vocal tract.
- Prosodic Features [59]: are used to model the speaking style of a speaker.
- High-level Features [52]: are used to model the lexicon of a speaker.

While all these features encode different speaker-dependent speech characteristics, their relative utility in performing speaker recognition depends on several factors. Short-term spectral features, for example, are extracted from short speech segments to model the vocal tract of the speaker efficiently. While these features are effective in clean speech scenarios, they are not robust to audio degradations [66]. Prosodic features, on the other hand, are derived from longer speech segments like syllables, words, and utterances to capture the speaking style of a speaker efficiently [102]. While prosodic features are relatively robust to audio degradations, they typically underperform the short-term spectral features in low-noise scenarios [102]. Therefore, the choice between different types of speech features can be based on the application scenario.

1.2.3 Effects of audio degradations on speech characteristics

Humans are efficient in performing speaker recognition in the presence of unknown types of audio degradations, also referred to as speaker recognition in mismatched noise conditions. In comparison, the MFCC feature, which is based on human auditory processing, struggles to perform in such scenarios [176]. Motivated by this, the authors in [176] propose the Gammatone Filterbank as an alternative to the Mel-filterbank for modeling the human auditory system. Compared to the Mel-filterbank, the Gammatone Filterbank has finer resolution at lower frequencies, which is claimed to better represent the human auditory model [58] and thus is a suitable modification to the MFCC feature extraction process. Additionally, the authors also replaced the logarithmic nonlinearity with the cubic root, to further improve the robustness of the features against audio degradations. This new proposed feature set was called Gammatone Frequency Cepstral Coefficients (GFCC) [176]. Another key reason for the poor performance of the MFCC features was identified to be the absence of any form of environmental compensation in the feature extraction process [87]. The authors in [1], hence, proposed noise-robust speech features called Power Normalized Cepstral Coefficients (PNCC) that incorporated a noise-suppression algorithm for suppressing the background excitation. Similar to the GFCC feature, PNCC also used Gammatone Filterbank instead of Mel-filterbank for extracting voice characteristics.

The MFCC feature extraction process disregards the phase information in the speech data, and the features are extracted only from the amplitude spectrum. The initial motivation behind disregarding the phase information was based on human auditory system experiments [58] In these experiments, the short-term phase spectrum did not provide enough performance benefits to justify the additional computational complexity of extracting phase-based features. However, recent studies [116, 127] have reported comparable and complementary speaker recognition performance of amplitude-based and phase-based features [121]. One of the recent works [141] used the Hilbert transform for combining the amplitude and phase information in speech data to generate a noise-robust and unified feature representation called the Mean Hilbert Envelope Coefficient (MHEC). Similar to the GFCC and PNCC, the MHEC feature extraction process also used the Gammatone Filterbank. The Hilbert envelope of the Gammatone Filterbank outputs is used to compute the MHEC features.

1.2.4 Speaker modeling

Speaker recognition is often categorized into text-dependent and text-independent methods. Text-dependent speaker recognition relies on the utterance of a fixed pass-phrase to recognize the user. Text-independent speaker recognition, in contrast, does not require any such fixed pass-phrase and can recognize the user from their vocal characteristics alone. In the scope of this work, we develop text-independent speaker recognition models robust to a wide variety of audio degradations.

One of the simplest text-independent speaker recognition models is the Vector Quantization (VQ) model, also known as the centroid model [32]. In the VQ model, the speech characteristics (such as the MFCC) from different speakers are formed into separate non-overlapping clusters. The centroids of these clusters also referred to as the codebook, represent the corresponding speaker templates. Any input probe sample is compared against the codebook to assign it to its closest matching speaker identity. The assumption of non-overlapping clusters in the VQ model was later relaxed to include overlapping clusters in the widely-popular Gaussian Mixture Model (GMM) based approach [135]. In this approach, instead of exclusively assigning a feature vector to a

single cluster, the feature vector is instead assigned a nonzero probability of originating from each cluster. The GMM based approach was further extended to the application of speaker verification using background speaker normalization [136]. The authors presented a technique for selecting a set of speakers to form a background speaker model. This model gave an estimate of the spread of speech features across the selected set of speakers.

In another work by Campbell et al., kernelized SVMs were combined with the GMM based approach [34]. The authors used speaker-adapted GMMs [137] to form a GMM-supervector. The GMM-supervector was then used to map an input speech utterance to a high dimensional feature space to derive kernels for the SVM. These high dimensional GMM-supervectors often require sophisticated matchers. Therefore, in order to enable the use of simple metrics, such as cosine similarity, for speaker verification, the high-dimensional GMM-supervectors were transformed to a lower-dimensional space, called the total variability space, using factor analysis [49]. A given speech utterance in this total variability space is represented by a low-dimensional vector called i-vector [50]. The i-vectors are then used for speaker verification.

Speaker recognition, like many other classification tasks, has attracted the application of deep learning-based techniques [105] for improving the current state-of-art. Richardson et al. [139] used spectral audio features (like MFCC) for performing speaker recognition on the input frame using deep neural networks (DNN). Zhang et al. [175] trained multi-layer perceptrons and Deep Belief Networks for learning discriminative feature transformations. Deep learning techniques, however, require large amounts of training data. Since large amounts of data are not always readily available, Richardson et al. [138] mixed synthetically generated noisy data along with the available clean speech data to train a denoising DNN that could be used as a front-end processing technique for speaker recognition. In one of the recent works [38], authors allude to the possibility of extracting glottal features for performing speaker recognition and have shown results for the same using their proposed 1D CNN based speaker recognition algorithm. In another work [154], a DNN based feature embedding, called xVector, was learned from MFCC features and combined with a PLDA classifier for performing speaker recognition. While the xVector technique is shown to outperform

the iVector-PLDA baseline on several public datasets, its performance on severely degraded data can not be ascertained. Also, due to the usage of a fully-connected DNN based architecture, the xVector model has almost 4.2 million learnable parameters, which necessitates the availability of a large amount of training data.

1.2.5 Effect of audio degradations on speaker modeling

Audio data captured in unconstrained scenarios are mostly noisy. Speaker recognition research over the last decade has, therefore, focused on developing noise-robust speaker recognition methods. Additive background noise is one of the most common types of noise found in speech signals. A popular technique commonly used to deal with the effects of background noise is spectral subtraction [28]. The robust feature estimation method [167], for example, preprocesses noisy speech utterances using spectral-subtraction to capture vocal source and vocal tract characteristics reliably. While preprocessing the noisy speech to obtain near clean speech is one way of dealing with degraded audio signals, another approach [114] relies on training a classifier on noisy data to make it robust to audio degradations. Such an approach can work well when the amount and type of audio-degradations are constrained. However, often there is an extensive amount of audio degradations, rendering portions of the audio unreliable for performing speaker recognition. The work in [107] combines missing data recognition with UBM-GMM to marginalize unreliable feature values and perform speaker recognition in noisy speech audio.

Both the iVector-PLDA [50] and UBM-GMM [137] based speaker recognition techniques use MFCC features for representing the speech data. However, it is important to note that the MFCC feature set is not robust [66] to audio degradations and can, therefore, potentially affect the performance of UBM-GMM and iVector-PLDA in noisy scenarios. Authors in [66] have, hence, used the Linear Predictive Cepstral Coefficients (LPCC) features, which are more robust to audio perturbations, for narrowing down the search space, and then used the MFCC features for performing speaker recognition in the reduced search space.

In order to alleviate the problems caused by degraded audio, several spectrum estimation meth-

ods and speech enhancement techniques have also been evaluated as front-end processing techniques for developing robust speaker recognition methods. Voice activity detection is one such technique used for detecting parts of the audio with speech activity in them. Sadjadi et al. [142] used it as a front-end processing technique for detecting and removing non-speech parts of the audio, which are typically long noisy audio segments.

Apart from additive noise, audio samples captured in indoor scenarios also suffer from convolutive reverberations. The work in [177] addresses this issue in a two-staged approach. It first uses the noisy speech data to train a DNN classifier to produce a binary time-frequency (T-F) mask. The mask identifies and segregates the unreliable T-F units at each audio frame. The masked output audio is then evaluated using GMM-UBM speaker models, trained in reverberant environments, to perform speaker recognition. Another problem associated with speech production in noisy environments is the Lombard effect [69], where the speakers involuntarily tend to adjust their vocal effort in-order to accommodate the noisy environment. This additional vocal effort applied by speakers perturbs their natural voice characteristics, thereby adversely affecting speaker recognition performance. Authors in [69] have further established the dependence of Lombard speech on noise type and noise level using a GMM based Lombard speech type classifier.

1.2.6 Challenges in speaker recognition

Speaker recognition, like face and fingerprint recognition, faces a large variety of challenges, as illustrated in Figure 1.5, which makes it a difficult problem. We discuss a few of the most fundamental problems in modeling speaker-dependent characteristics for performing speaker recognition, below:

- There are inherent differences in the recognizability of different speakers. The famous work on 'Doddington Zoo,' explained these differences by categorizing all speakers into four broad categories based on their behavior concerning automatic recognition systems [53]. For example, speakers who are particularly challenging to model were termed 'Goats,' 'Sheeps' on the other hand, are easy to model and comprise the majority of the population. 'Lambs'

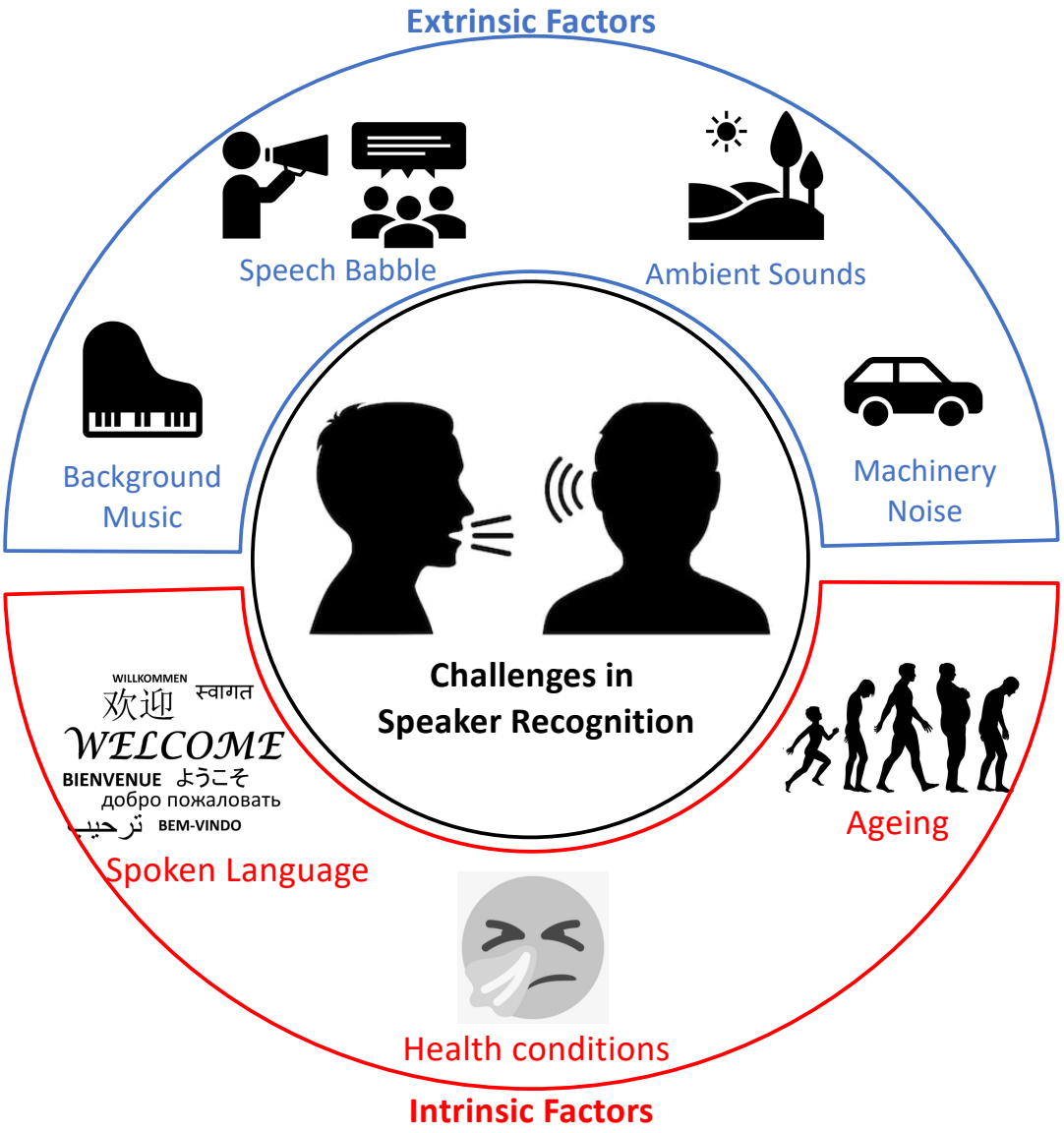


Figure 1.5: An illustration of different challenges in speaker recognition.

represent speakers who are easy to imitate, while 'Wolves' are particularly adept at imitating other speakers. The differences in the gender [96, 104] and the spoken-language [61, 99] exacerbates the differences in the recognizably of different speakers. Therefore, it is essential to develop models that offer equitable speaker recognition performance across a large and diverse set of speakers varying in their gender and languages.

- Audio degradations introduced at different steps of audio recording, transmission, and storage further worsen the problem of speaker modeling [97]. Furthermore, a mismatch in the

acoustic environments of the training and evaluation data severely degrades the performance of a speaker recognition model [101, 155, 156]. Therefore, it is imperative to develop noise-robust speaker recognition models that can operate reliably across a wide variety of audio degradations.

- The reliability of voice features extracted from speech audio of short duration depends directly on the amount of usable speech data within. The presence of audio degradations in the speech audio further reduces the amount of usable speech data in the audio sample [172]. Therefore, it is essential to develop speech feature extraction and embedding algorithms that reliably extract speaker-dependent speech features from short-duration speech audios.

- A majority of the speaker recognition algorithms still rely on hand-crafted features such as the MFCC and the LPC for performing speaker recognition. While such features are effective in clean speech scenarios, they are not robust to audio degradations [40, 66]. Therefore, it is important to develop data-driven techniques for automatically extracting speaker-dependent speech features that adapt to a wide variety of acoustic environments and are robust to audio degradations.

- While most of the state-of-the-art speaker recognition techniques attempt to model the physical characteristics of the the voice modality to perform automatic speaker recognition, the behavioral characteristics of human voice, such as the prosody, are often not accounted for in the development of speaker recognition systems. Prosodic features capture non-segmental aspects of speech such as the intonation, speaking style, accent, and pronunciation of the speaker. However, unlike short-term spectral features, prosodic features are extracted from longer segments of speech and thus necessitate availability of long duration speech data ($>$ 3 mins per sample). Therefore, it is important to collect speech data that capture prosodic features and develop algorithms that can leverage the prosodic features for aiding the performance of speaker recognition algorithm that rely only on the physical traits of human voice.

1.3 Thesis Contributions

In this thesis, we propose several deep learning-based techniques for performing robust speaker recognition from audio samples collected in diverse acoustic environments. Consequently, we address some of the challenges presented in the previous section. The major contributions of this thesis are listed below.

1. A convolutional neural network (CNN) based on 1D filters, rather than 2D filters, has been developed for extracting noise-robust speech embedding from cepstral speech features, such as the Mel-frequency Cepstral Coefficients (MFCC). The filters in the CNN are designed to learn inter-dependency between cepstral coefficients extracted from audio frames of fixed temporal expanse. Also, the CNN is designed to extract speech embeddings independently from each input audio frame and retain only the embeddings that are common across several input audio frames. Such an approach is shown to reliably extract noise-robust speech embeddings as it focuses on extracting speaker-dependent speech features that are consistent across different frames and thus can deal with varying audio degradations across the frames.

2. Further, we approach the problem of speaker recognition from severely degraded audio data by judiciously combining two commonly used speech features: Mel Frequency Cepstral Coefficients (MFCC) and Linear Predictive Coding (LPC). Our hypothesis rests on the observation that MFCC and LPC capture two distinct aspects of speech, viz. speech perception and speech production. A carefully crafted 1D Triplet Convolutional Neural Network (1D-Triplet-CNN) is used to combine these two features in a novel manner, thereby enhancing the performance of speaker recognition in challenging scenarios. Extensive evaluation on multiple datasets, different types of audio degradations, multi-lingual speech, and varying length of audio samples convey the efficacy of the proposed approach over existing state-of-the-art speaker recognition methods.

3. Automatic speaker recognition algorithms, traditionally, use pre-defined filterbanks, such as Mel-Frequency and Gammatone filterbanks, for characterizing speech audio. The design

of these filterbanks is based on domain-knowledge and limited empirical observations. The resultant features, therefore, may not generalize well to different types of audio degradation. We propose a deep learning-based technique to induce the design of a filterbank from vast amounts of speech audio. The purpose of such a filterbank is to extract features that are robust to degradations in the input audio. To this effect, a 1D convolutional neural network (1D-CNN) is designed to learn a time-domain filterbank called DeepVOX directly from raw speech audio. Secondly, an adaptive triplet mining technique is developed to efficiently mine the data samples best suited to train the filterbank. Thirdly, a detailed ablation study of the DeepVOX filterbanks reveals the presence of both vocal source and vocal tract characteristics in the extracted features. Experimental results on VOXCeleb2, NIST SRE 2008 and 2010, and Fisher speech datasets demonstrate the efficacy of the DeepVOX features across a variety of audio degradations, multi-lingual speech data, and varying-duration speech audio. The DeepVOX features also improve the performance of existing speaker recognition algorithms, such as the xVector-PLDA and the iVector-PLDA.

4. Automatic speaker recognition algorithms typically characterize speech audio using short-term spectral features, such as MFCC and LPC, that encode the physiological and anatomical aspects of speech production. Such algorithms do not fully capitalize on speaker-dependent characteristics present in behavioral speech features. In this work, we propose a prosody encoding network called DeepTalk for extracting vocal style features directly from raw audio data. The DeepTalk method outperforms several state-of-the-art speaker recognition systems across multiple challenging datasets. The speaker recognition performance is further improved by combining DeepTalk with a state-of-the-art physiological speech feature-based speaker recognition system. We also integrate DeepTalk into a current state-of-the-art speech synthesizer to generate synthetic speech. A detailed analysis of the synthetic speech shows that the DeepTalk captures F0 contours essential for vocal style modeling. Furthermore, DeepTalk-based synthetic speech is shown to be almost indistinguishable from real speech in the context of speaker recognition.

5. A text-independent speaker recognition system relies on successfully encoding speech factors such as vocal pitch, intensity, and timbre to achieve good performance. A majority of such systems are trained and evaluated using spoken voice or everyday conversational voice data. Spoken voice, however, exhibits a limited range of possible speaker dynamics, thus constraining the utility of the derived speaker recognition models. Singing voice, on the other hand, covers a broader range of vocal and ambient factors and can, therefore, be used to evaluate the robustness of a speaker recognition system. However, a majority of existing speaker recognition datasets only focus on the spoken voice. In comparison, there is a significant shortage of labeled singing voice data suitable for speaker recognition research. To address this issue, we assemble *JukeBox* - a speaker recognition dataset with multilingual singing voice audio annotated with singer identity, gender, and language labels. We use the current state-of-the-art methods to demonstrate the difficulty of performing speaker recognition on singing voice using models trained on spoken voice alone. We also evaluate the effect of gender and language on speaker recognition performance, both in spoken and singing voice data.

6. Speaker recognition systems often rely on ideal audio conditions, such as minimal background noise and neutral speaking style, to achieve good performance. However, practical application scenarios frequently deviate from ideal conditions, reducing speaker recognition performance. The singing audio is an example that combines the intrinsic factors of speech variability in speaking style with the extrinsic elements of background music. We first extend a publicly available singing voice dataset, JukeBox, with corresponding speaking voice data, and refer to it as JukeBox-V2. We then study the effect of variations in audio conditions between the speaking and singing voice on speaker recognition performance. Next, we propose using domain adaptation for developing speaker recognition methods robust to varying speaking styles and audio conditions. For example, in the JukeBox-V2 dataset, for the domain-adapted 1D-Triplet-CNN method, the true match rate at a false match rate of 1% improves by over 12% and 2% for the singing and spoken voice, respectively. Finally, a

detailed analysis of the domain-adapted method's speech embeddings explains its generalizability across varying speaking styles and audio conditions.

The above contributions are discussed in detail in the following chapters.

CHAPTER 2

SPEAKER RECOGNITION USING ONE DIMENSIONAL CONVOLUTIONAL NETWORKS

Portions of this chapter appeared in the following publication:

Chowdhury, Anurag, and Arun Ross. "Extracting sub-glottal and supra-glottal features from MFCC using convolutional neural networks for speaker identification in degraded audio signals." In International Joint Conference on Biometrics (IJCB), pp. 608-617. IEEE, 2017.

2.1 Introduction

In a typical speaker recognition system, an input voice sample is either: identified as one of the enrolled speakers or verified against a claimed identity. However, in an unconstrained scenario, the voice sample often contains audio degradations, such as background noise and channel distortions, consequently degrading the speaker recognition performance. The detrimental effect of such audio degradations on the speaker recognition performance varies with the type and amount of degradation present in the input speech audio. For example, the background noise of machinery is relatively easier to circumvent due to its static nature, however, the dynamic nature of background speech babble (from other humans) makes it extremely disruptive to the speaker recognition system.

Traditionally, speaker recognition systems characterise an input speech audio by analysing its constituent frequencies. This analysis is typically done using frequency-domain filterbanks, such as the Mel-filterbank, that are designed to extract important speaker dependent speech characteristics. However, in the case of degraded audio samples, the constituent frequencies of the input speech audio also contains components of noise. Depending upon the type and amount of noise present in the speech audio, certain frequency components of the input speech can be completely obscured by the co-existing noise components. This renders the aforementioned frequency components useless for the task of speaker recognition. While prior knowledge of the spectral constitution

of the background noise can be used to mitigate its detrimental effect to a certain extent [177], such a solution cannot be generalized to unknown noise profiles. However, the automatic feature learning capability of data-driven approaches, such as the convolutional neural networks, can be used to identify the frequency bands that are least affected the audio degradations. A non-linear combination of the corresponding noise-robust frequency responses can then be used to perform speaker recognition in degraded audio signals.

In this chapter, we discuss a deep learning-based algorithm for automatically learning the subset of filters whose frequency bands are least affected by the audio degradations. The corresponding filter responses are then used to extract noise-robust speaker-dependent speech characteristics to perform speaker recognition from degraded audio signals. We use the commonly employed Mel-Frequency Cepstral Coefficients (MFCC) for representing the audio signals. A convolutional neural network (CNN) based on 1D filters, rather than 2D filters, is then designed. The filters in the CNN are designed to learn inter-dependency between cepstral coefficients (filter-bank responses) extracted from audio frames of fixed temporal expanse. Our approach aims at extracting noise-robust speaker-dependent speech characteristics from the cepstral coefficients extracted from degraded audio signals. The performance of the proposed method is compared against existing baseline schemes on both synthetically and naturally corrupted speech data. Experiments convey the efficacy of the proposed architecture for speaker recognition.

2.2 Rationale behind automatic speaker recognition

For performing automatic speaker recognition, it is important to first understand how human speech is generated at the source. For generating voiced speech sounds, the sound source is provided by periodic vibration of the vocal folds by a process known as *phonation*. For phonation to occur, the ratio of the air pressure below the glottis (*sub-glottal*) to air pressure above the glottis (*supra-glottal*) must exceed a certain positive value [4]. The shape and size of the vocal tract imparts individuality to a speaker's voice characteristics. MFCC features, as discussed further in the section 2.3.3.1, have been extensively used for capturing acoustic features of human vocal tract,

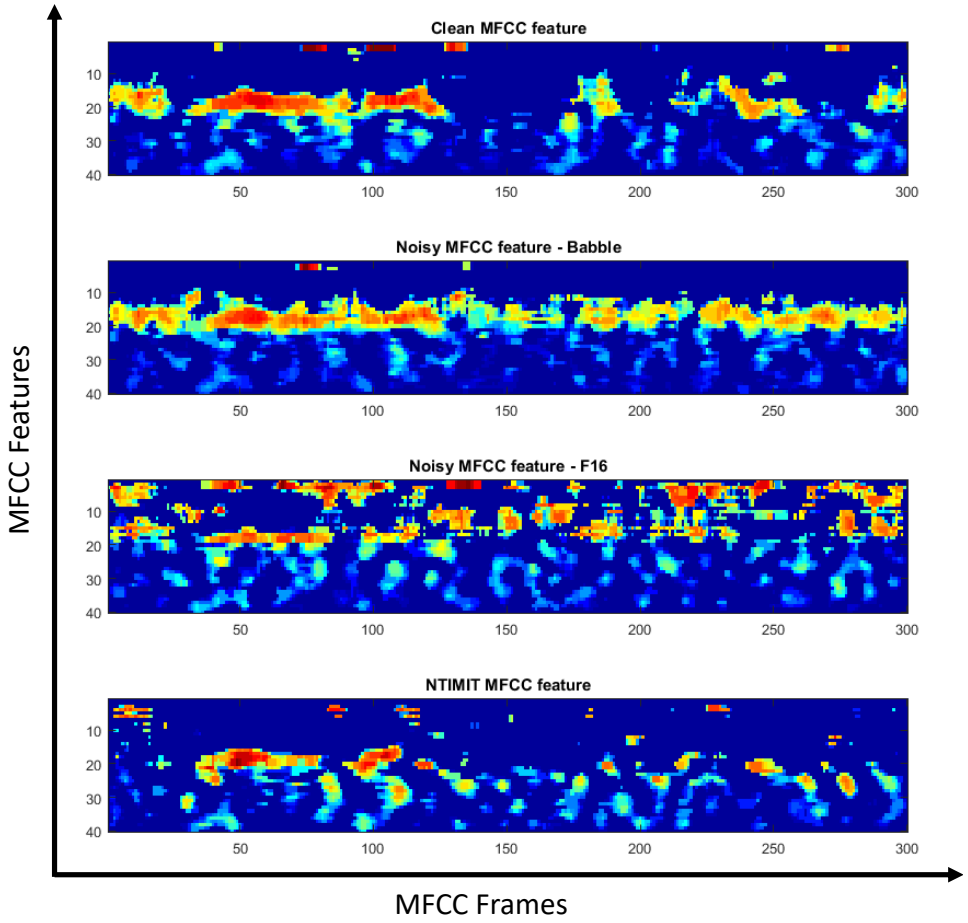


Figure 2.1: Visual representation of MFCC feature strip of a clean audio clip - TIMIT (first row); corresponding noisy audio clip - Babble (second row); noisy audio clip - F16 (third row); and noisy audio clip - NTIMIT (fourth row).

which we have incorporated in our approach to perform speaker recognition.

Our approach for solving the problem of speaker recognition uses a Convolutional Neural Network (CNN) uniquely designed to learn the speaker dependent characteristics from patches of MFFC audio features. The MFCC features are widely used in the speech and speaker recognition community as they represent the shape of the envelope of the power spectral density of the speech audio, which in turn is a manifestation of the shape of the human vocal tract.

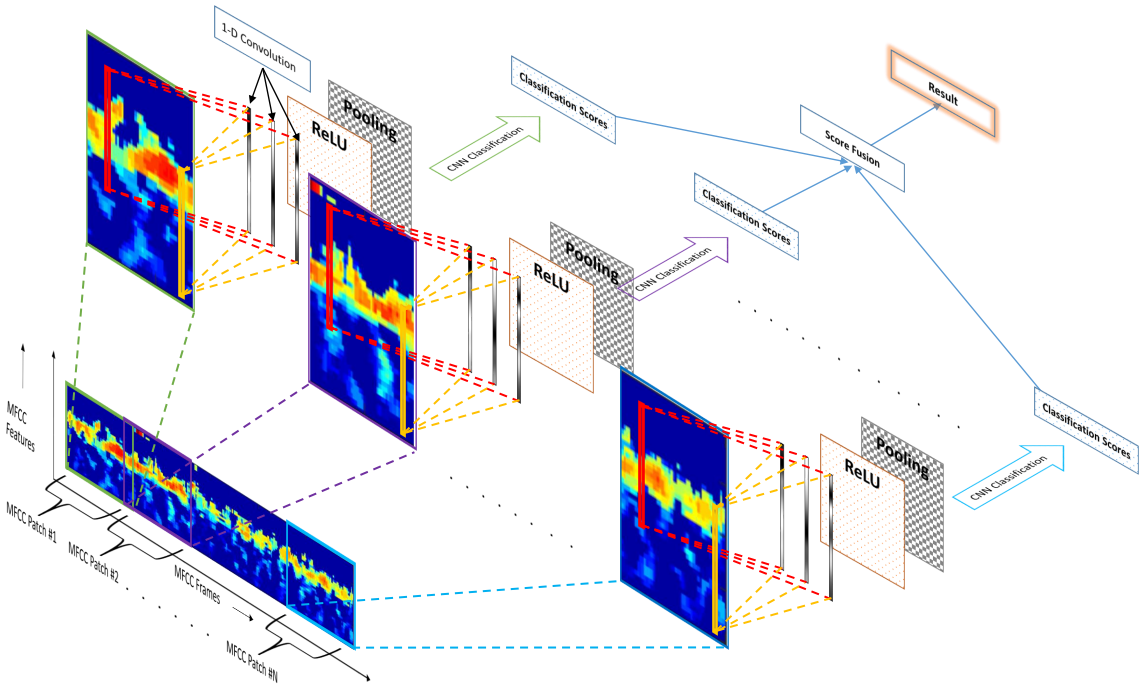


Figure 2.2: An illustration of the proposed speaker identification algorithm using 1-D CNN. The input MFCC feature strip is split into MFCC patches and evaluated on the trained CNN. The classification scores from different patches are fused to arrive at a classification decision.

2.3 1-Dimensional Convolutional Neural Network (1-D CNN) based Speaker Recognition

In the proposed work, we use 1-D convolutional filters for learning speaker dependent features from MFCC features for performing speaker identification in degraded audio signals. We model the problem of speaker identification as an image classification problem and propose a CNN architecture that is uniquely suited for speech data analysis and works particularly well for the task of speaker identification in degraded audio signals.

2.3.1 Speech Parametrization

MFCC features are very popular in the speech and speaker recognition community. A detailed account of the MFCC feature extraction process can be found in [133, 135]. We used the VOICEBOX [29] toolbox for extracting MFCC feature from the audio data. Our 40 dimensional MFCC feature vector comprises of 20 mel-cepstral coefficients that includes the zeroth order coefficient,

and 20 first order delta co-efficients. The hamming window is used in the time domain and triangular filters are used in the mel-domain.

2.3.2 Data Organisation

The input audio clip is split into smaller clips, of fixed temporal expanse, called audio *frames*. The number of audio frames in the input audio clip is determined by the length of a frame and the frame stride. The length of an audio frame, n, is a function of the sampling frequency, fs. In the VOICEBOX [29] toolbox, n is expressed as follows:

$$n = 2^{\lfloor log_2(0.03 * fs) \rfloor}. \tag{2.3.1}$$

The frame stride is chosen to be $n/2$. We extract 40-dimensional MFCC features per audio frame of an audio clip. Upon extracting the MFCC feature from an audio clip, we obtain a two dimensional feature matrix, which is referred to as MFCC *feature strip* in this work. Each MFCC feature strip is of size $40 \times F$, where F is the number of extracted frames. Since the length of the input audio could be of arbitrary length, we extract MFCC feature *patches* containing fixed number of audio frames from the MFCC feature strip of the audio clip. The patches are extracted using a moving window approach, where the size of the window is set to 200 frames and the stride value to 100 frames. A visual representation of the MFCC feature strip of a clean audio sample and its corresponding noisy versions can be seen in Figure 2.1. The MFCC feature patches in the training and test sets were modified by subtracting the corresponding average image from them, in order to zero-center the data. The modified MFCC feature patch of size 40×200 is now used as a two dimensional data input to the CNN network architecture described below.

2.3.3 1-D Convolution

A traditional CNN architecture consists of a sequence of layers. Each layer transforms the input data by applying layer specific operations on the input and passing it over to the next layer. The three most common layer types found in a CNN architecture are: Convolutional Layer, Pooling

Layer and Fully-Connected Layer. The convolutional layer in a CNN is where majority of the learning process takes place. Design and placement of the filters along the various layers of a CNN determine the “concepts" that are learned at each layer.

Deciding the shape of filters in CNNs is crucial to effectively learning the target concept from the input data. As discussed in [98], small square shaped filters are especially good for learning local patterns in image data, such as edges and corners, due to the high correlation between pixels in a small local neighborhood. However, that is not the case in the context of MFCC feature strips, as there is no local semantic structure (to our knowledge) that can be captured by a 2-D filter. As represented in Figure 2.1, the pixel values along Y axis corresponding to the MFCC features are on a logarithmic scale, while the pixel values along X axis corresponding to the time domain are on a linear scale. Hence a 1-D filter is better at learning speaker dependent characteristics from the MFCC features placed along the Y axis.

2.3.3.1 Sub-glottal and Supra-glottal features

In the field of speech recognition, 1-dimensional filters across the time variable have shown promising results [174] by effectively learning temporal characteristics in the data. However, in the context of text independent speaker recognition, the temporal relevance of speaker dependent characteristics across MFCC feature frames is greatly reduced (but not eliminated), as the content of the speech has often no bearing on the identity of the speaker (especially in cases where the data is collected in a controlled lab environment rather than in a natural conversational mode). Hence, learning to extract acoustic speech features that are speaker dependent and text independent, like supra-glottal and sub-glottal resonances [33], are more beneficial for the task of speaker identification.

Features of the sub-glottal and supra-glottal vocal tract capture the acoustics of the tracheabronchial airways and are known to be noise robust for speaker identification [66]. MFCC features, in-turn, are known to capture acoustics of the supra-glottal and sub-glottal vocal tract [21]. These features have been reliably estimated from MFCC features [22], indicating the potential of learning

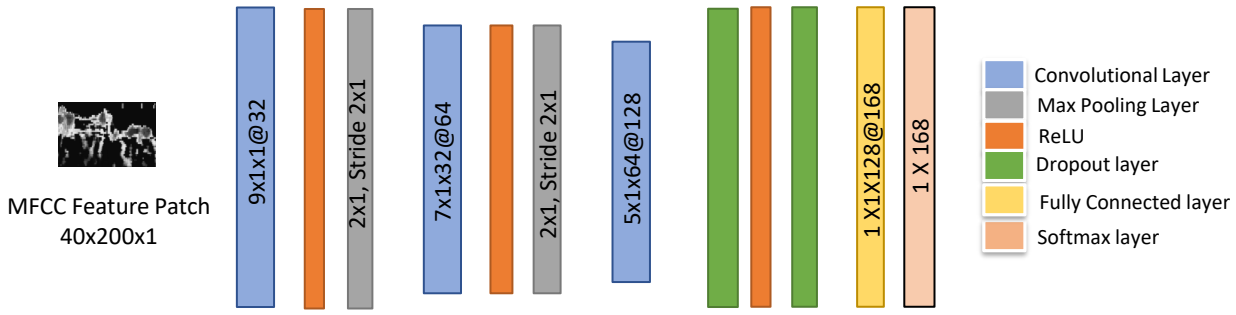


Figure 2.3: Architecture of the CNN used for Speaker Identification from degraded audio samples. The input is a $40 \times 200 \times 1$ MFCC feature patch to the CNN. The last layer gives a classification score to each of the 168 speakers in the testing set in the TIMIT and NTIMIT datasets.

and extracting such noise-robust speaker dependent acoustic features from MFCC feature patches.

The design of our CNN architecture is motivated by the intent to learn and extract such speaker dependent acoustic features from MFCC feature patches for speaker identification. Such acoustic features are usually stable only for a short-period of time, say 20ms, which is effectively captured by the MFCC feature extraction process. Hence, we design 1-dimensional convolutional filters of various sizes aligned along the Y axis, as illustrated in Figure 2.2, in order to glean the acoustic features resident in mel-cepstral frequency coefficients. The final architecture of our CNN is presented in Figure 2.3.

2.3.4 ReLU NonLinearity and Pooling layers

The filter responses from each of the convolutional layelrs are made to pass through ReLU nonlinearity as, unlike sigmoid activation functions, they do not suffer from the problem of vanishing gradients. Further, we used max-pooling to reduce the size of the parameter space to be learnt by the network.

2.3.5 Dropout

Dropout layers were added to introduce regularization in the CNN being trained. It provides the dual benefit of making the CNN robust towards perturbations in the input data while also mitigating the problem of over-fitting to the training data.

Table 2.1: Identification Results on the SITW, NTIMIT and Noisy variants of TIMIT speech dataset.

Exp. #	Training set	Testing Set	Accuracy (Rank 1 in %)			Accuracy (Rank 5 in %)		
			UBM-GMM	i-vector-PLDA	1-D CNN	UBM-GMM	i-vector-PLDA	1-D CNN
1	Babble, F16, R1,V1	Car, Factory, R2, V2	3.86	1.98	32.93	15.57	8.53	65.57
2	Car, Factory, R2, V2	Babble, F16, R1,V1	9.52	10.61	35.61	21.52	29.26	67.95
3	Babble, Car, R2, V2	F16, Factory, R1, V1	9.22	14.08	47.61	18.55	31.64	75.09
4	F16, Factory, R1, V1	Babble, Car, R2, V2	6.84	4.86	38.59	20.13	14.08	66.96
5	Car, F16, R1, V1	Babble, Factory, R2, V2	6.25	3.27	21.13	15.37	11.01	47.81
6	Babble, Factory, R2, V2	Car, F16, R1, V1	20.03	10.61	24.60	34.42	31.15	50.99
7	NTIMIT	NTIMIT	52.38	57.14	62.50	81.54	87.5	85.71
8	SITW	SITW	70	49.44	71.11	86.11	73.33	83.33

Table 2.2: Identification Results on the Noisy variants of TIMIT speech dataset in presence of the extended gallery-set ($1052 + 168$ speakers). The extended gallery consists of audio samples from the Fisher speech dataset also.

Exp. #	Training set	Testing Set	Accuracy (Rank 1 in %)			Accuracy (Rank 5 in %)		
			UBM-GMM	i-vector-PLDA	1-D CNN	UBM-GMM	i-vector-PLDA	1-D CNN
1	Babble, F16, R1,V1	Car, Factory, R2, V2	1.58	1.09	13.78	9.92	5.95	35.31
2	Car, Factory, R2, V2	Babble, F16, R1,V1	1.09	2.87	46.03	2.97	5.75	69.94
3	Babble, Car, R2, V2	F16, Factory, R1, V1	1.78	5.15	39.68	3.47	13.59	65.57
4	F16, Factory, R1, V1	Babble, Car, R2, V2	1.88	0.99	37.00	11.30	4.86	57.73
5	Car, F16, R1, V1	Babble, Factory, R2, V2	0	0.19	24.50	0	0.39	51.19
6	Babble, Factory, R2, V2	Car, F16, R1, V1	16.56	6.54	51.98	26.19	19.14	72.42

2.3.6 Score level fusion and Decision

In the testing phase, as illustrated in the Figure 2.2, the input MFCC feature strip, X, is split into MFCC patches, $x_i, i\epsilon\{1, 2, 3, ..., N\}$, where, N, is the number of patches. For every input MFCC patch, x_i, the CNN gives a set of classification scores, $\{s_{i,j}\}, j\epsilon\{1, 2, 3, ..., C\}$, corresponding to the C speakers (e.g., $C = 168$ in the TIMIT and NTIMIT test datasets). Here, $s_{i,j}$, is the classification score assigned to the j^{th} speaker for the i^{th} patch.

Scores from all the patches extracted from the audio clip are then added to give fused classification scores, $\{S_j\}$, for the entire audio clip:

$$S_j = \sum_{i=1}^{N} s_{i,j}, \forall j.$$

The input audio is then assigned to the speaker j^* where,

$$j^* = \underset{j}{\operatorname{argmax}}\{S_j\}.$$

2.4 Experiments

2.4.1 Datasets

We used the TIMIT [60] Acoustic-Phonetic Continuous Speech Corpus, NTIMIT [80], SITW [110] and Fisher [44] datasets to demonstrate the performance of our algorithm for text-independent speaker recognition under degraded conditions.

2.4.1.1 TIMIT Dataset

The TIMIT dataset provides clean speech recordings of 630 speakers. There are 462 speakers in the training set and 168 speakers in the testing set. The dataset contains of eight major dialects of American English. There are ten sessions of 3 seconds each (so 10 audio samples) per speaker in the dataset. The text spoken by the speakers in the training set and test set are disjoint, making the speaker recognition experiments text-independent.

In our experiments, TIMIT dataset was perturbed [20,74] with synthetic noise of different types (given below) from the NOISEX-92 [165] noise dataset. The noisy versions of the TIMIT dataset were generated in simulated room environments with different acoustic properties and reverberation levels, thereby introducing convoluted reverberations into the noise profile. The synthetically generated noisy datasets have the following noise characteristics:

1. Noise Type: Following four types of noises were added to the TIMIT dataset:
    1.1. F-16: Noise generated by engine of F-16 fighter aircraft.
    1.2. Babble: Noise generated by rapid and continuous background human speech.
    1.3. Car: Noise generated by engine of a car.
    1.4. Factory: Noise generated by heavy machinery operating in a factory environment.
2. Signal to Noise Ratio (SNR): The resultant noisy datasets were each generated at three different SNR levels, viz., 20 dB, 10dB and 0dB.

3. Room Size: The noisy dataset were generated in a simulated room environment with two different room sizes (4m and 20m, side length of cube), referred to as R1 and R2 in the protocol.

4. Reverberation: Two different reverberation coefficients were used to introduce additional noise in the data, referred to as V1 and V2 in the protocol.

2.4.1.2 Fisher English Training Speech Part 1 dataset

The Fisher English Training Speech Part 1 Speech dataset contains conversational speech data collected over telephone channels between pairs of speakers. This dataset has over $12,000$ speakers. Conversations pertaining to a subset of $1,052$ speakers from the Fisher dataset were chosen for the experiments in this work. Audio pertaining to each speaker in the conversation is then segmented out and processed with voice activity detection to remove empty audio segments from the audio. The audio of each speaker was then split into smaller audio snippets of around 3-second duration each. We extract 60 audio snippets for each speaker from their conversational audio.

2.4.1.3 NTIMIT Dataset

NTIMIT [80] dataset consists of speech from the TIMIT dataset that was transmitted and re-collected over a telephone network. The speech content and speakers in the NTIMIT dataset are identical to that of the TIMIT dataset. But since the NTIMIT is collected over a telephone network, it has noise characteristics inherent to the telephone channel, thereby resulting in a noisy version of the TIMIT dataset. Even though the average SNR of NTIMIT dataset is higher (36dB) than that of the noisy versions of the TIMIT dataset that we had created (section 3.4.2.1), the former provides a much more realistic noise profile.

2.4.1.4 Speakers in the Wild (SITW) Database

The Speakers in the Wild (SITW) dataset [110] contains speech samples collected from open-source media for benchmarking and evaluating text-independent speaker recognition algorithms. Since the SITW data was not collected in a controlled setting, it contains real noise, reverberation, intra-speaker variability and compression artifacts. There are 299 speakers in the dataset (119 in the training set and 180 in the testing set) with variable number of audio samples of differing lengths per speaker. Audio of each speaker from the dataset is processed with voice activity detection to remove any empty audio segments. The audio for each speaker was then split into smaller audio snippets of around 3-second duration each. We extract 10 audio snippets for each speaker from their conversational audio.

2.4.2 Experimental Protocols

In the experiments involving noisy variants of the TIMIT dataset, we ensure disjoint noise characteristics in the training and testing sets as shown in Table 2.1. For example, in experiment 1, the training set consists of audio samples that are simulated to be recorded in a room of size R1 and reverberation coefficient V1, with additive background noise of type “Babble" and “F16".

Apart from the six experiments on the noisy TIMIT datasets, we also perform speaker identification experiments on the NTIMIT and SITW datasets. The training and the testing sets in the NTIMIT dataset share the same noise profile (that of telephone channels), unlike the disjoint noise profiles in the noisy versions of TIMIT dataset created by us. The noise content in the SITW datset varies greatly over samples both within and between different speakers.

Additionally, we also extended the six experiments on the noisy TIMIT datasets by adopting an extended gallery set comprising of a subset of 1052 speakers from the Fisher dataset alongside the original 168 speakers in the testing set of the TIMIT dataset. The extended gallery set, therefore, has 1220 speakers.

2.4.2.1 UBM-GMM [137] based Speaker Identification

To obtain baseline performance on the eight experiments laid out in Table 2.1, we train a Universal Background Model (UBM) [137] using data from the speakers in the training set. The trained UBM is then adapted using data from the speakers in the test set, to obtained speaker-adapted GMM models. For adapting the UBM to individual speakers, nine audio samples per speaker is used, and the remaining audio sample per speaker is reserved for testing.

2.4.2.2 i-vector-PLDA [63] based Speaker Identification

To obtain a second baseline performance on the eight experiments laid out in Table 2.1, we train an i-vector-PLDA based speaker recognition system as implemented in the MSR identity toolkit [143]. Similar to the protocol for the UBM-GMM experiment, we use nine audio samples per speaker from the testing set for adapting the i-vector models, and the remaining audio sample per speaker is reserved for evaluation.

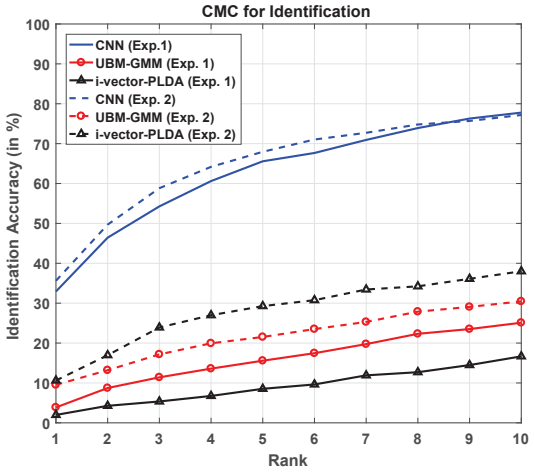
2.4.2.3 1-D CNN based Speaker Identification

The eight experiments, given in Table 2.1, were then conducted using the proposed 1-D CNN based Speaker Identification algorithm. Since the CNN based algorithm does not require a background model unlike UBM-GMM [137], we directly train the CNN on the speakers in the test set, with nine audio samples per speaker. The remaining audio sample per speaker is used in the test set.

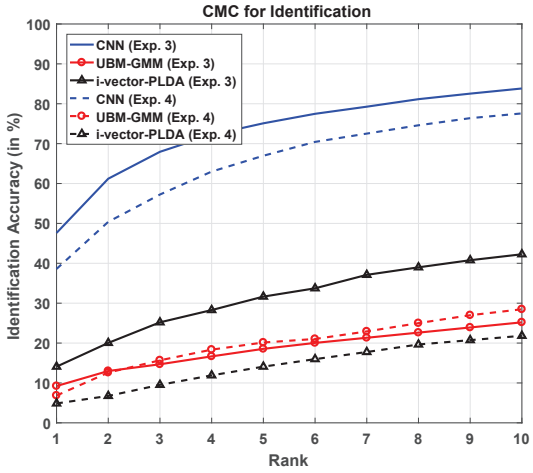
2.4.2.4 Extended Gallery Speaker Identification

The six experiments, given in Table 2.2, are the extended gallery experiments that were done to test the discriminative power of the algorithms in presence of an extended gallery set. The speaker recognition models in the six extended-gallery experiments were trained in exactly the same way as they were done for the first six experiments in Table 2.1. The gallery set of 168 speakers from the TIMIT dataset are augmented with a subset of 1052 speakers from the Fisher English Training

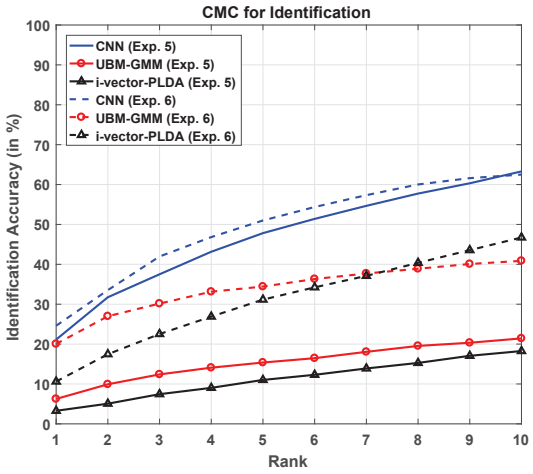
Speech Part 1 Speech dataset. The probe data is sourced from only the 168 speakers in the TIMIT dataset. Therefore, for each probe sample, the algorithms now have to make a decision from a pool of 1220 speakers, where 168 are from the TIMIT dataset and 1052 are from the Fisher dataset.



(a) Experiment 1 & 2



(b) Experiment 3 & 4

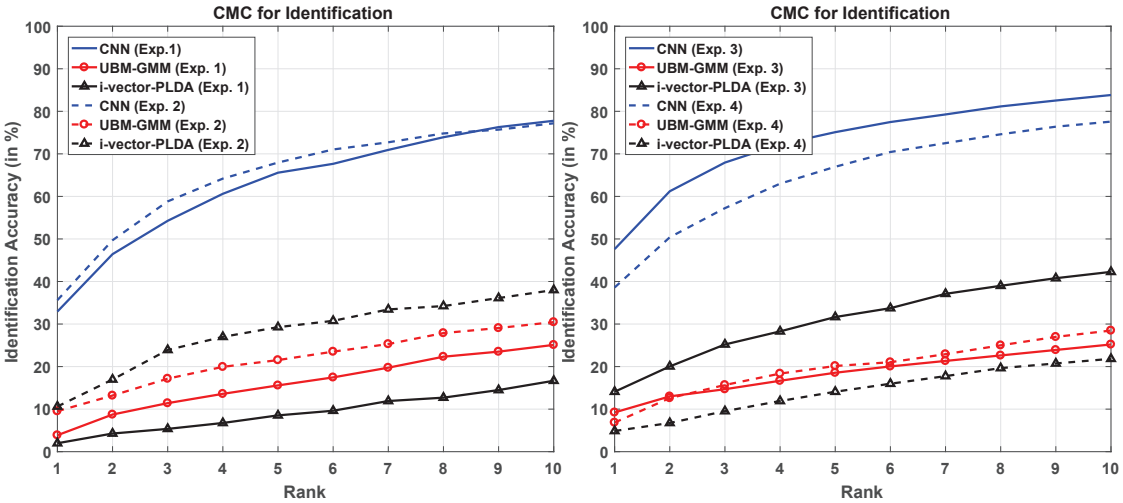


(c) Experiment 5 & 6

Figure 2.4: CMC curves for the speaker identification experiments on the noisy variants of the TIMIT dataset (Exp. 1 to 6) using UBM-GMM, i-vector-PLDA and 1-D CNN algorithms.

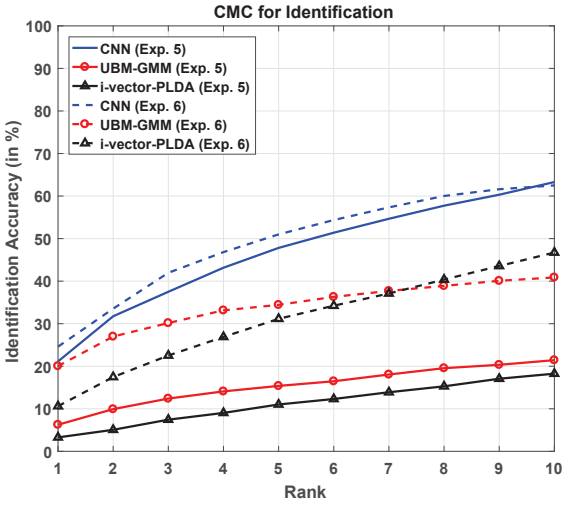
2.5 Results and Analysis

The results of the identification experiments are given in Tables 2.1 and 2.2. Both Rank-1 and Rank-5 identification accuracies (in %) are reported for the baseline methods and the proposed



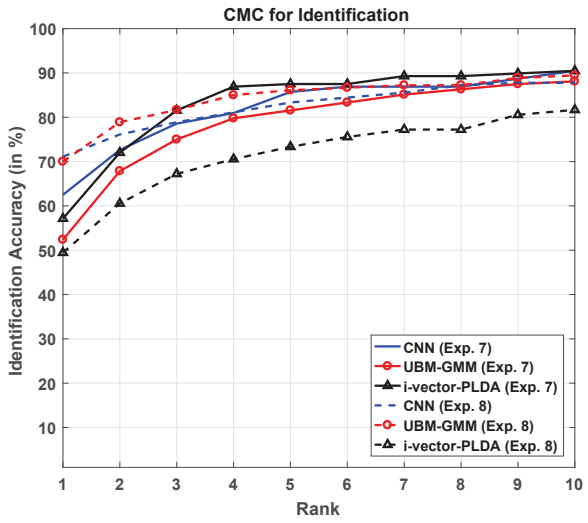
(a) Experiment 1 & 2

(b) Experiment 3 & 4



(c) Experiment 5 & 6

Figure 2.5: CMC curves for speaker identification experiments in the presence of extended gallery-set ($1052 + 168$ speakers) on the noisy variants of the TIMIT dataset (Exp. 1 to 6) using UBM-GMM, i-vector-PLDA and 1-D CNN algorithms.



(a) Experiment 7 & 8

Figure 2.6: CMC curves for the speaker identification experiments on the NTIMIT (Exp. 7) and SITW (Exp. 8) dataset using UBM-GMM, i-vector-PLDA and 1-D CNN algorithms.

method. The Cumulative Match Characteristic (CMC) curves are given in Figures 2.4 and 2.6.

- The identification accuracy of the 1-D CNN based speaker identification algorithm is vastly superior at Rank 1 across all eight experiments in Table 2.1.

- The average identification accuracy across the first six experiments on the noisy TIMIT datasets is **33.40**% at Rank 1 and **62.40**% at Rank 5 for 1-D CNN, 9.29% at Rank 1 and 20.92% at Rank 5 for UBM-GMM and 7.56% at Rank 1 and 20.94% at Rank 5 for i-vector-PLDA.

- In the experiments on NTIMIT dataset, it is important to note that i-vector-PLDA outperforms UBM-GMM at both Rank 1 and Rank 5 indices, and it also outperforms the proposed 1-D CNN based algorithm at Rank 5. This could be attributed to the fact that i-vector-PLDA outperforms UBM-GMM in low noise scenarios and, since the NTIMIT dataset has higher average SNR (36dB) compared to that of the noisy variants of TIMIT dataset (10dB), the i-vector-plda performs better on the NTIMIT dataset. Even though the i-vector-PLDA outperforms 1-D CNN at Rank 5, it should be noted that 1-D CNN significantly outperforms i-vector-PLDA at Rank 1.

- In the SITW dataset, 1-D CNN based algorithm modestly outperforms the baseline algorithms at Rank 1.

- In the extended gallery experiments, the accuracy of 1-D CNN based speaker identification algorithm continues to be superior at both Rank 1 and Rank 5 indices across all six experiments. It is noteworthy that in experiment 5, UBM-GMM has a 0% accuracy at both Rank 1 and Rank 5, as it completely failed to identify the correct speakers at lower ranks in the extended gallery set. This substantiates the challenges of performing speaker identification in large datasets.

- On average, across the first six experiments in Table 2.1, UBM-GMM, i-vector-PLDA and 1-D CNN correctly identify the same 0.14% of the test samples at Rank 1. 1-D CNN correctly identifies an additional 26.60% of the test samples over both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1. However, the 1-D CNN based algorithm fails to correctly identify 2.64% of the test samples that were correctly identified by both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1.

- In the seventh experiment in Table 2.1, on the NTIMIT dataset, UBM-GMM, i-vector-PLDA and 1-D CNN based algorithms correctly identify the same 41% of the test samples at Rank 1. The 1-D CNN based algorithm correctly identifies an additional 10% of the test samples over both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1. However, 1-D CNN based algorithm fails to correctly identify 11% of the test samples that were correctly identified by both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1.

- In the eigth experiment in Table 2.1, on the SITW dataset, UBM-GMM, i-vector-PLDA and 1-D CNN based algorithms correctly identify the same 41.11% of the test samples at Rank 1. The 1-D CNN based algorithm correctly identifies an additional 0.06% of the test samples over both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1. However, the 1-D CNN based algorithm fails to correctly identify 0.02% of the test samples that were correctly identified by both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1.

- For the experiments with the extended gallery set in Table 2.2, on average, all three algorithms, UBM-GMM, i-vector-PLDA and 1-D CNN, correctly identified the same 0.82% of the test samples at Rank 1. 1-D CNN correctly identifies an additional 31.46% of the test samples over both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1. However, the 1-D CNN based algorithm fails to correctly identify 0.16% of the test samples that were correctly identified by both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1. This establishes the superior discriminative power of the 1-D CNN based algorithm over both the baseline algorithms.

- In both the baseline algorithms and proposed algorithm, the MFCC features are used as input; but the performance of the 1-D CNN vastly improves over that of the baselines. This suggests that the 1-D CNN is better at extracting important speaker dependent characteristics, like sub-glottal and supra-glottal features, in presence of audio degradations.

2.6 Summary

Degradations in speech audio can distort and mask the speaker dependent characteristics in the audio signal. Traditional speaker identification approaches like UBM-GMM and i-vector-PLDA fail to perform well in noisy scenarios. The 1-D CNN-based speaker recognition algorithm is robust to a wide range of audio degradations as evidenced in the experimental results, but it still fails to correctly identify more than 60% of the samples at Rank 1 across the six experiments on noisy variants of the TIMIT dataset. This brings to focus the challenges of the task and the scope for improvement.

The current algorithm is developed and evaluated for an *identification* setting; in the next chapter, we will address the speaker *verification* task on severely degraded audio signals. We will also discuss the theoretical background of speech production and speech perception in humans and incorporate the relevant domain-knowledge in our speaker verification system.

CHAPTER 3

FUSING MFCC AND LPC FEATURES USING 1D TRIPLET CNN FOR SPEAKER RECOGNITION IN SEVERELY DEGRADED AUDIO SIGNALS

Portions of this chapter appeared in the following publication:

Chowdhury, Anurag, and Arun Ross. "Fusing MFCC and LPC Features using 1D Triplet CNN for Speaker Recognition in Severely Degraded Audio Signals." IEEE Transactions on Information Forensics and Security (2020).

3.1 Introduction

In the previous chapter, we introduced a one-dimensional convolutional network (1D-CNN) based approach for performing speaker recognition from degraded audio signals. In this chapter we extend the technique of 1D-CNN for fusing voice perception features (MFCC) and voice production features (LPC) for performing speaker recognition in severely degraded audio signals.

Speaker recognition algorithms are negatively impacted by the quality of the input speech signal. In this chapter, we approach the problem of speaker recognition from severely degraded audio data by judiciously combining two commonly used features: Mel Frequency Cepstral Coefficients (MFCC) and Linear Predictive Coding (LPC). Our hypothesis rests on the observation that MFCC and LPC capture two distinct aspects of speech, viz., speech perception and speech production. A carefully crafted 1D Triplet Convolutional Neural Network (1D-Triplet-CNN) is used to combine these two features in a novel manner, thereby enhancing the performance of speaker recognition in challenging scenarios. Extensive evaluation on multiple datasets, different types of audio degradations, multi-lingual speech, varying length of audio samples, etc. convey the efficacy of the proposed approach over existing speaker recognition methods, including those based on iVector and xVector.

The performance of speaker recognition systems is adversely impacted by a number of fac-

tors. For example, noisy environments and animated conversations involving multiple subjects can confound a speaker recognition system. Further, the quality of the microphone and distance of the subject from the microphone can also lead to a marked drop in speaker recognition accuracy. In forensic applications, the audio signal may be severely degraded leading to difficulties in recognizing individuals.

In this chapter, we aim to develop a speaker recognition algorithm that is robust to a wide range of audio capture quality, ambience and perturbations. Some of the current voice recognition enabled products [5] already incorporate advanced hardware based measures, such as circular arrays of far-field microphones, for enabling robust audio input interfaces, thereby aiding their speech and speaker recognition capabilities. Our goal, on the other hand, is to develop a software based solution that is not restricted to specific audio interfaces for performing robust speaker recognition.

Impact of audio degradation on speaker recognition in real life scenarios

Most speaker recognition enabled products are vulnerable to challenging audio conditions such as low audio SNR, differing dialects and accents, and background noise [15]. The problem is exacerbated in presence of background noise [15]. One of the most challenging daily-life scenario for digital assistants is the babble noise [93] in crowded environments, such as coffee shops. Babble noise typically comprises speech from multiple individuals in the background. In this case, the voice commands from the intended user can be misinterpreted or even go unattended due to lack of usable audio data in the presence of extensive background noise.

In the next section, we will discuss some of the more established speaker recognition algorithms which help define the performance and utility of speaker recognition in real life scenarios. A majority of speaker recognition algorithms rely on some form of short-term spectral speech features like Mel Frequency Cepstral Coefficients (MFCC) and Linear Prediction Cepstral Coefficient (LPCC). However, their reliance on one type of speech feature constrains their performance and utility. For example, while MFCC features are known to represent perceptual speech, they are also unreliable in presence of audio degradations. This reduces the performance and reliability of

algorithms based on MFCC features alone.

This is where we position our work in relation to the current existing literature, as detailed in the following sections. We propose a deep learning based algorithm, referred to as 1D-Triplet-CNN, to combine speech *perception* features and speech *production* features, given by MFCC and LPC features, respectively, for learning a joint feature space that efficiently models the entire *speech chain*. Finally, we describe the speaker verification experiments conducted in this work to demonstrate the superior representation capability of the joint feature space, even in the presence of varying types and strength of background noise.

In the following sections, we will discuss the proposed technique we have designed for speaker recognition on considerably degraded speech data.

3.2 Theoretical Foundations

Text-independent speaker recognition can be seen as the process of extracting the speaker dependent characteristics from the human speech, regardless of the textual content within, for uniquely identifying the human speaker at the source. This process can be well described by using the speech chain [51] which expounds the physics and biology of the spoken language in an ordered fashion.

3.2.1 Speech Chain

The speech chain, illustrated in Figure 3.1, is typically used for explaining the physics and biology involved during formulation, articulation, propagation and reception of a message from a speaker to the listener. While the focus in speech chain at its extremities is on the message being transferred through the chain, it is also important to notice the change in information rate of the message as it passes through the chain. The information rate contained in the transmitted spoken message is significantly higher than the base information rate of the text message itself, as also shown in Figure 3.1. The 'Neuro-Muscular controls' in the speech production process encodes the pronunciation elements of the message as articulations and the 'Vocal Tract System' generates the

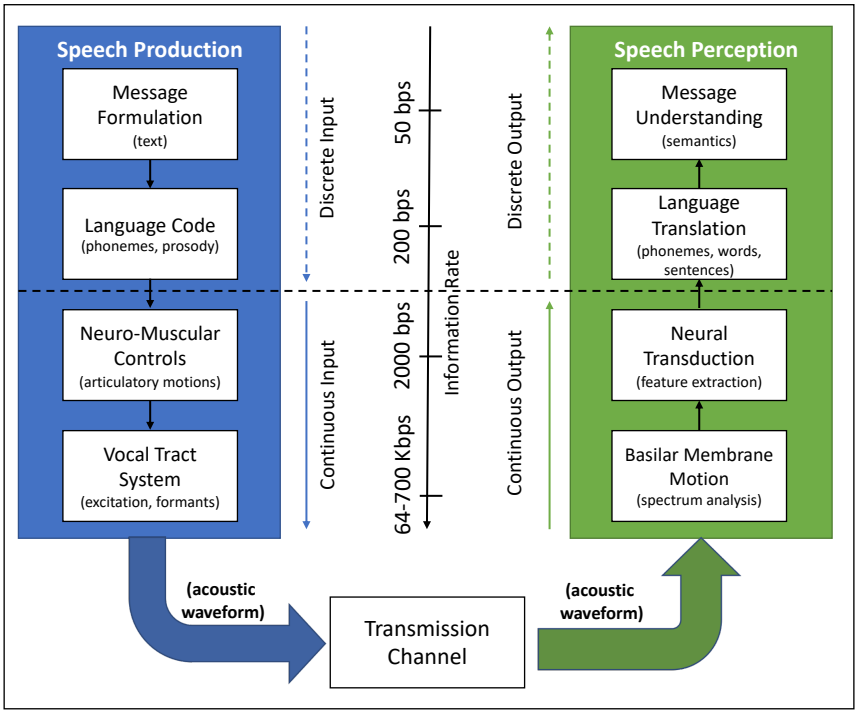


Figure 3.1: A visual representation of the speech chain as given in [67]

sound from the articulation hence imparting the spoken language its acoustic properties. Thus, the articulatory and acoustic information in the speech leads to increase in the net information content of the speech. The speech perception process on other hand performs spectral analysis on the audio transmitted through the channel using the 'Basilar Membrane Motion' which is further passed through the 'Neural Transduction' phase to extract speech features essential for performing tasks like speech, speaker and language recognition.

From the perspective of speaker recognition, our interest is focused on two different parts of the speech chain. First being the 'Vocal Tract System' in the 'Speech production' process as it imparts the speech its acoustic properties and can be used for modeling the vocal tract system that gives an individual their unique voice characteristics. We use Linear Predictive Coding (LPC) for accomplishing this task and is elucidated upon in the upcoming sections. Second being the 'Neural Transduction' phase in 'Speech perception' process where the speech features, as perceived by the listener, are extracted from the speech audio and hence can be used for modeling the human auditory system that can discriminate between the speakers in the speech audio. We use Mel-Frequency Cepstral Coefficients (MFCC) for accomplishing this task and is further explained upon

in the upcoming sections.

3.2.2 Vocal tract modeling using Linear Predictive Coding (LPC)

According to the source-filter model [113] of speech, human voice can be seen as filter (vocal tract) output of excitation from an energy source (lungs). Vocal tract of humans can be modeled as a time-varying digital filter. Furthermore, the all-pole model of filter design is chosen for easier estimation and analysis of the human vocal tract. Thus, the transfer function of the digital filter equivalent of vocal tract can be given by,

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}}. \tag{3.2.1}$$

Speech data is sequential in nature and, for modeling the vocal tract, we assume that the voice acoustics of the n^{th} speech sample ($S[n]$) can be viewed as a combination of p past speech samples. Thus, the n^{th} speech sample ($S[n]$) can be written as:

$$S[n] = \sum_{k=1}^{p} \alpha_k S[n-k] + G.u[n], \tag{3.2.2}$$

where, $S[n-k], k = 1, 2, 3 \ldots p$ are the p past speech samples, G is the gain factor, $u[n]$ is the excitation corresponding to the n^{th} speech sample, α_k's are the vocal tract filter coefficients, and "." represents the scalar multiplication operation. *Linear Predictive Coding* (LPC) is often referred to as "inverse filtering" as its aim is to determine the "all zero filter" which is the inverse of the vocal tract model. Just as in the source-filter model of human voice the LPC model also estimates voice acoustics of the n^{th} speech sample $\hat{S}[n]$, conditioned on previous speech samples, given as,

$$\hat{S}[n] = \sum_{k=1}^{p} \hat{\alpha}_k S[n-k]. \tag{3.2.3}$$

where α_k's are the LPC parameters. The error in prediction is given by,

$$e[n] = S[n] - \hat{S}[n].kl \tag{3.2.4}$$

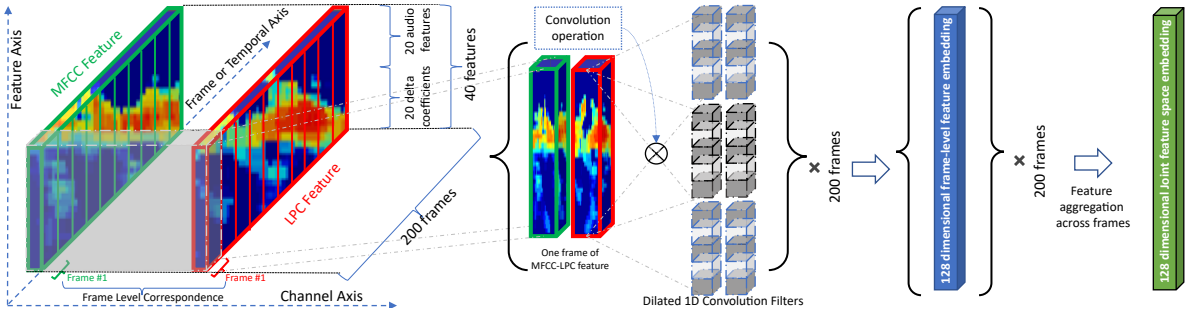


Figure 3.2: A visual representation of feature fusion in the proposed 1D-Triplet-CNN architecture

$$E = \sum_{n=1}^{N} (e(n))^2 \tag{3.2.5}$$

Minimizing the above energy (E), using auto-regressive modelling [56], will help obtain the Inverse-Vocal Tract filter model. The above energy can be minimized using auto-regressive modelling [56] for obtaining the Inverse-Vocal Tract filter model. The filter coefficients of the Inverse-Vocal Tract filter model are the LPC model parameters (α_k), which provide an estimate of the human vocal tract filter coefficients.

3.2.3 Perceptual speech features using Mel-Frequency Cepstral Coefficients (MFCC)

Humans are exceptionally good at identifying other *known* humans from their voice [68], even in presence in audio degradations. In order to model the human auditory perception system, it is important to understand how humans recognize speakers from their voice. Human auditory system is made up of outer ear, middle ear and inner ear. Our interests are more vested in the inner ear because this is where the auditory nerves pick up sound signals from the cochlea and deliver it to the brain. The human cochlea is a part of inner ear and its function is to separate sounds based on their frequency content and transduce the sound waves to electrical signals. This part of the auditory processing is done in 'Basilar Membrane Motion' phase in the speech chain, as given in Figure 3.1. The auditory nerve fibers then carry these electrical signals to the brain. Our brain then performs complex spectral and temporal processing of the sound signal, as shown in the 'Neural Transduction' phase in the speech chain in Figure 3.1, for extracting speech features. These features are then used for performing tasks like speech, speaker, and language recognition.

Mel-cepstral frequency coefficients (MFCC) have been used in literature [120], for modeling the human auditory perception system. MFCC feature is also a very popular choice for performing speaker recognition. Therefore, we also chose MFCC for representing the perceptual speech features in a given audio sample for our work. In the coming sections, we will discuss the MFCC feature extraction process in detail and will also elucidate upon the MFCC and LPC feature fusion and the deep learning architecture that we propose for speaker verification.

3.2.4 Rationale behind fusing LPC and MFCC features for speaker recognition

As discussed in the previous sections both LPC and MFCC model different characteristics of a speaker's voice which could be used separately for speaker recognition. However, the nature of the voice features being captured by LPC and MFCC are complimentary, as the MFCC features describe the perceptual speech features while the LPC features describe the vocal tract model for the speaker in a speech audio. The combination of MFCC and LPC features, therefore, uniquely represent the voice characteristics of a speaker. In this work, we devise a CNN based feature level fusion algorithm that combines and projects the voice characteristics in the MFCC and LPC feature spaces into a d-dimensional joint feature space. Here, the value of d depends on the CNN architecture. The joint feature space is learnt in such a way (as explained in later sections) that the joint feature representation captures highly discriminative speaker dependent voice characteristics for improving speaker recognition performance.

3.3 Proposed 1D-Triplet-CNN for performing speaker recognition

We aim to do speaker verification from a fusion of LPC and MFCC features and therefore propose our own *1D-Triplet-CNN* architecture, given in Figure 7.1, for the task. In our network design we create three clones of a 1D-CNN model, forming the proposed 1D-Triplet-CNN. The weight matrices of the three clones share the same memory space and are called 'shared weights'. During training, an update made to any of the three 1D-CNN clones updates the 'shared weights' and is, therefore, reflected across the entire triplet of CNNs. For training the network, we provide a data triplet D_t as input to the CNN, given by $D_t = (S_a, S_p, S_n)$. Here, S_a and S_p, called

anchor and *positive* samples respectively, are two different speech samples taken from a subject 'A', whereas, S_n, the negative sample, is a speech sample from another subject 'B', such that $A \neq B$.The task of the loss function in the training phase, described in section 4.2.1.5, is to help the network learn the similarity between the anchor sample and the positive sample and the dissimilarity between the anchor sample and the negative sample.

In the testing phase, as shown in Figure 7.1, we arrange the trained CNN into a Siamese network instead of a triplet network. Unlike the training phase, here we only need two copies of the trained CNN for matching a data pair $D_p = (S_1, S_2)$. Here,S_1 and S_2 are audio samples from two different recordings. The two copies of the trained CNN are then used to extract embeddings for S_1 and S_2, individually. Cosine similarity metric is used to compare the extracted embeddings and provide a corresponding match score. In an ideal case, embedding of a sample pair belonging to same subject should give a match score close to 1, while embeddings of a sample pair belonging to two different subjects should result in a match score close to -1.

In the following sections we will discuss our network design choices and the thought process behind it.

3.3.0.1 Speech Parametrization and Data Organization

Similar to [38], we split the input audio clips into smaller patches, called *audio frames*, using a sliding window of length $\lfloor 0.02 * fs \rfloor$ and stride $0.5 * \lfloor 0.02 * fs \rfloor$. Here, fs is the sampling frequency of the input audio. Same window sizes and strides were used for both LPC and MFCC extraction process. We use voice activity detection (VAD), prior to feature extraction, to remove unvoiced portions from the input audio. VOICEBOX [29] toolbox is used for extracting 40 dimensional LPC and MFCC *feature frames* from the audio frames. Each LPC feature frame comprises of 20 LPC coefficients concatenated with 20 first order delta coefficients. Similarly, MFCC feature frame comprises of 20 mel-cepstral coefficients(including zeroth order coefficient) and 20 first order delta coefficients. The extracted features were also normalized using cepstral mean variance

normalization (CMVN). In our experiments, we are able to achieve better generalizability by using CMVN and hence is an important part of the algorithm. The number of audio frames that can be extracted from an audio depends on its sampling frequency and duration. Therefore, for training the CNN on input of fixed dimensionality, we randomly sample 200 contiguous feature frames, called *feature patch*, from every audio in every batch. Hence, each feature patch is of size 40×200. Doing this, we also achieve a form of data augmentation similar to 'random cropping' operation used in image based CNNs.

3.3.0.2 Feature Level Fusion of LPC and MFCC features

We stack the MFCC and LPC feature patches along the third dimension to create a, $40 \times 200 \times 2$ dimensional, two-channel feature patch referred to as *MFCC-LPC*. Here, the first channel corresponds to the MFCC feature patch and the second to the LPC patch. The two-channels of the MFCC-LPC feature are then combined in the proposed CNN architecture, as illustrated in Figure 3.2, using dilated 1D convolution filters. The proposed CNN architecture is designed to transform the 2-channel, 40-dimensional representation of each MFCC-LPC feature frame into a 128-channel, 1-dimensional frame-level feature embedding. Here, the 128 output channels represent a vector in the, hence learnt, 128-dimensional *Joint Feature Space*. The frame-level embeddings are aggregated across the 200 input frames, using average pooling, to output a 128 dimensional joint-feature space representing the speaker dependent information present in the input MFCC-LPC features.

3.3.0.3 Dilated 1D Convolutions

The design of convolutional layers in a CNN plays a fundamental role in determining its learning capability and efficiency. Each convolutional layer along the depth of a CNN learns different "concepts" from the data and transforms the data for layers further deeper in the CNN. As also mentioned by authors in [38], unlike images, speech data does not exhibit some of key properties that a CNN leverages for learning from image data. For example, pixels in images bear spatial

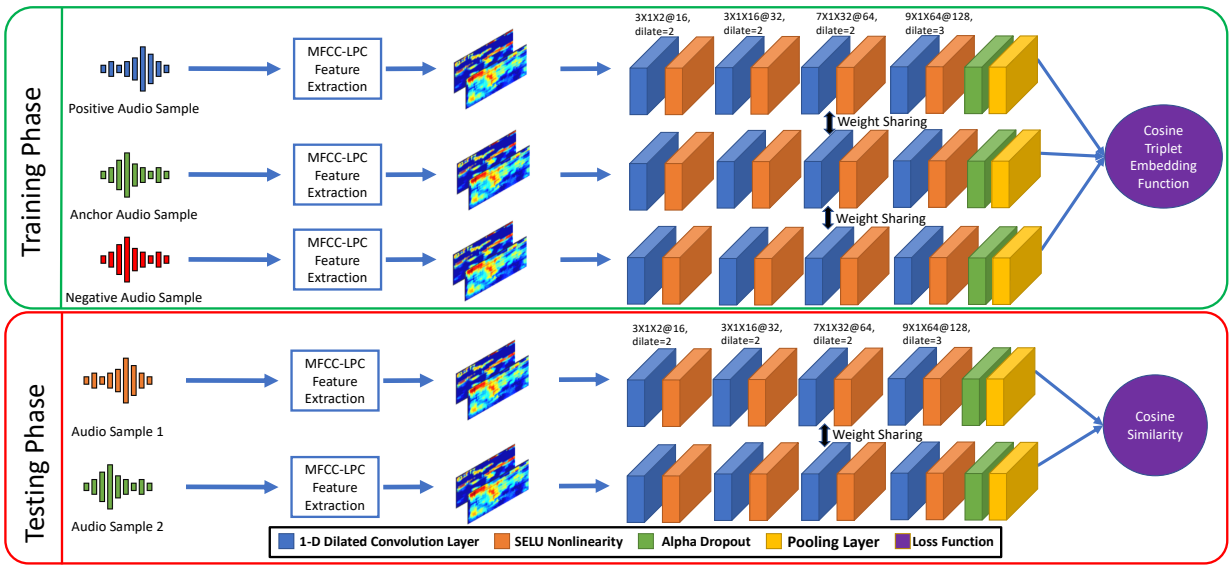


Figure 3.3: A visual representation of the proposed 1D-Triplet-CNN for performing speaker verification from degraded audio samples

relationships in a local neighborhood, in form of a semantic structure, which can be learnt by using 2-D convolutional filters in the CNN .

Speech data on other hand, when represented in form of two dimensional feature patches (e.g MFCC and LPC), does not exhibit similar semantic structures in a 2-D local neighborhood. The main reason behind this, as pointed in [38], is that the pixel values along Y axis correspond to the feature values (MFCC/LPC), which in case of MFCC features are placed on a logarithmic scale, while the pixel values along X axis vary with time on a linear scale. Therefore any semantic relationships that might exist should be constrained along 1D neighborhoods in the X and Y axes individually.

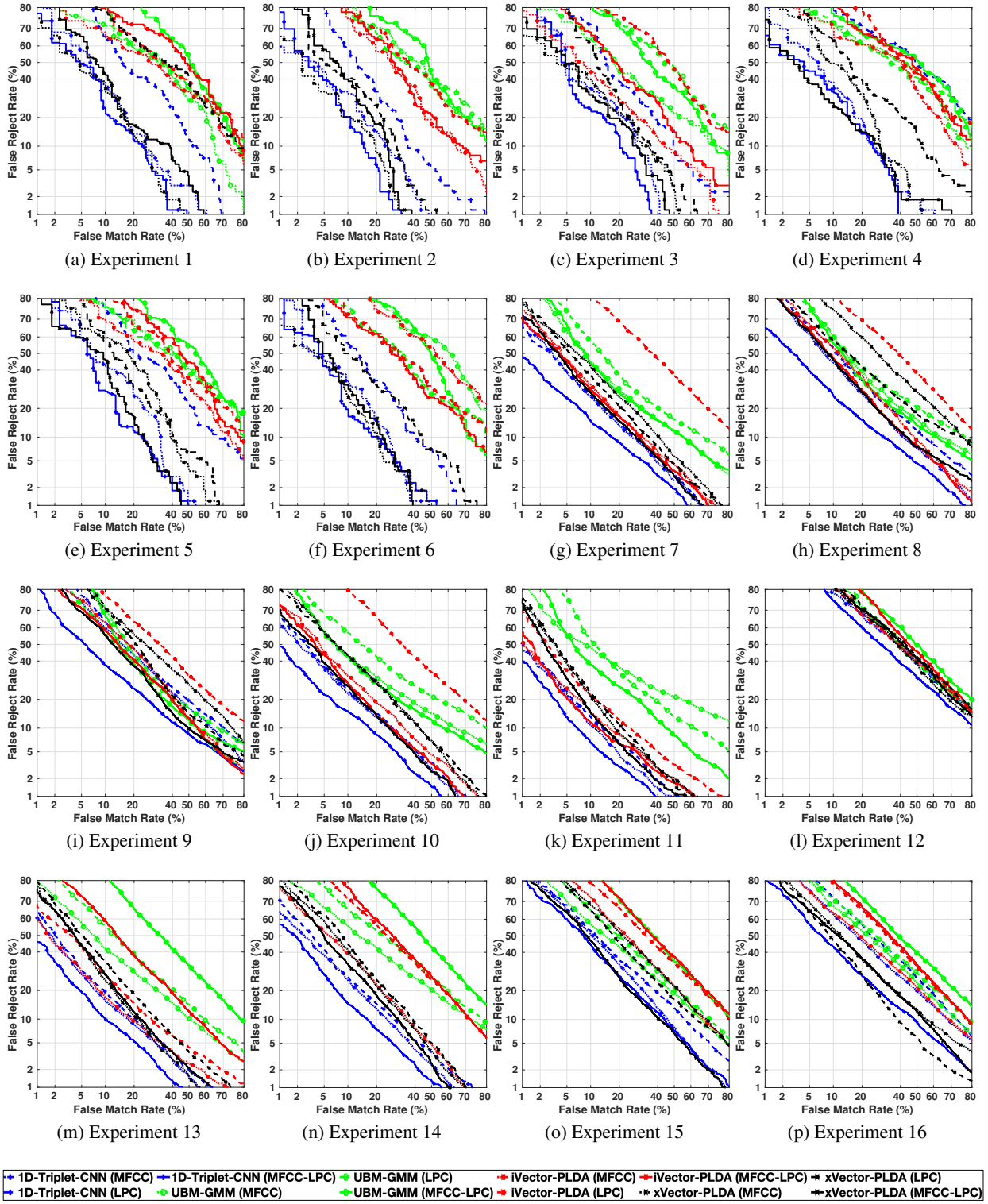
Another important point, to be noted, is that even though an audio signal is constantly changing, the speaker dependent voice characteristics are assumed to be stable only within short time scales (25ms). Therefore, we work on short term audio segments, called *audio frames*, as explained in the feature extraction process in section 3.3.0.1. The MFCC or LPC feature corresponding to that audio frame is called a *feature frame*. Thus, a feature frame extracted from an audio frame represents the audio properties of only that particular audio frame and does not bear any relationship with its neighboring frames in the context of speaker recognition. Putting these domain specific

constraints together we decide to use 1D convolutional filters along the feature dimension (Y axis), as introduced in [38], in our CNN for learning speaker dependent speech characteristics. We have further used *dilated* 1D convolutional layers instead of conventional 1D convolutional layers with 1D pooling as done in [38]. We have chosen to replace pooling operation with dilated convolutions, firstly, as an effort to minimize the data loss within the network due to pooling layers. Secondly, dilated convolutions also help in increasing the receptive field rapidly without greatly increasing the computational cost, as also done in [124]. This is because dilated convolutions, also known as convolution with holes, learn sparse filters. For example, a 1D-convolution filter of kernel size 5×1 with a unit dilation factor actually learns a sparse filter of size 9×1 with alternating indices populated. Such a filter, therefore, spans a larger receptive field than a traditional 1D-convolution filter of size 5×1, while using same number of parameters. Dilated 1D-convolutions can, therefore, replace pooling layers for increasing the receptive field in the network without sustaining any data loss introduced by the latter.

In our work, dilated convolutions are specifically beneficial over conventional convolutions followed by pooling for another reason: it allows the network to learn sparse relationships between the feature values within a feature frame. Learning such sparse relationships is particularly useful for learning speaker dependent features in degraded audios. Our intuition behind this is that in case of degraded audios containing structured background noise, the frequency bands closer to that of the noise are degraded uniformly in dense local regions. Hence, learning sparse features using *dilated* 1D convolutions prevents the network to learn from local dense regions and is, therefore, more robust to such audio degradations.

3.3.0.4 SELU Non-Linearity and Alpha Dropout

Similar to spectral subtraction, the authors in [38] subtracted data-mean from the training and validation datasets for zero centering their data. In our experiments, we found this to generalize poorly across datasets degraded with unknown types of noise profiles. This is primarily due to the fact that spectral subtraction based data normalization methods only work for *known* types of



(a) Experiment 1 (b) Experiment 2 (c) Experiment 3 (d) Experiment 4

(e) Experiment 5 (f) Experiment 6 (g) Experiment 7 (h) Experiment 8

(i) Experiment 9 (j) Experiment 10 (k) Experiment 11 (l) Experiment 12

(m) Experiment 13 (n) Experiment 14 (o) Experiment 15 (p) Experiment 16

Figure 3.4: DET curves for the speaker verification experiments on the degraded TIMIT dataset (Exp. 1 to 6), degraded Fisher dataset (Exp. 7 to 10) and, the clean and degraded NIST SRE 2008 and 2010 datasets (Exp. 11 to 16) using UBM-GMM, iVector-PLDA, xVector-PLDA and 1D-Triplet-CNN algorithms on MFCC, LPC and MFCC-LPC feature sets.

Table 3.1: Verification Results on the degraded TIMIT speech dataset.

Exp. #	Training set / Testing set	MFCC / LPC / MFCC-LPC							
		TMR@FMR=10%				minDCF($P_{tar} = 0.01$)			
		1D-Triplet-CNN	UBM-GMM	iVector-PLDA	xVector-PLDA	1D-Triplet-CNN	UBM-GMM	iVector-PLDA	xVector-PLDA
1	S1 / S2	61 / 34 / **75**	27 / 18 / 7	32 / 20 / 18	63 / 22 / 56	8.55 / 10 / **7.95**	9.57 / 9.46 / 9.94	8.63 / 10 / 9.16	**7.95** / 9.52 / 9.27
2	S2 / S1	67 / 42 / **79**	18 / 23 / 14	25 / 13 / 22	**79** / 58 / 60	6.54 / 9.7 / 7.48	9.82 / 9.52 / 9.82	9.34 / 9.82 / 9.4	**6.52** / 8.72 / 8.69
3	S3 / S4	65 / 27 / **76**	35 / 17 / 19	53 / 17 / 32	60 / 37 / 69	8.92 / 9.75 / 8.33	9.04 / 9.64 / 9.94	8.21 / 9.52 / 9.58	**7.19** / 9.4 / 8.14
4	S4 / S3	61 / 19 / 67	26 / 26 / 22	29 / 12 / 19	50 / 35 / **73**	8.38 / 9.64 / 7.18	9.64 / 9.16 / 9.16	8.92 / 10 / 9.88	8.86 / 9.93 / **7.07**
5	S5 / S6	56 / 24 / **71**	22 / 23 / 10	30 / 15 / 17	40 / 42 / 53	9.1 / 9.64 / 8.8	9.94 / 9.88 / 9.94	9.69 / 10 / 9.94	8.98 / 9.94 / **8.37**
6	S6 / S5	66 / 43 / **80**	22 / 23 / 17	36 / 27 / 29	73 / 50 / 68	7.77 / 10 / 7.6	9.04 / 9.58 / 9.46	9.28 / 9.87 / 9.04	**7.24** / 9.28 / 9.22

Exp. #	Training set / Testing set	MFCC / LPC / MFCC-LPC			
		Equal Error Rate (EER, in %)			
		1D-Triplet-CNN	UBM-GMM	iVector-PLDA	xVector-PLDA
1	S1 / S2	17 / 33 / **16**	41 / 42 / 50	38 / 42 / 49	17 / 45 / 17
2	S2 / S1	16 / 25 / **13**	45 / 44 / 48	36 / 43 / 36	14 / 21 / 18
3	S3 / S4	20 / 29 / **17**	32 / 48 / 41	23 / 47 / 32	21 / 22 / 19
4	S4 / S3	19 / 48 / **17**	47 / 44 / 50	44 / 46 / 47	23 / 29 / **17**
5	S5 / S6	23 / 37 / **14**	45 / 45 / 52	40 / 42 / 47	29 / 25 / 17
6	S6 / S5	18 / 26 / **14**	55/ 38 / 49	51/ 42 / 37	15 / 26 / 17

Data Subset	S1	S2	S3	S4	S5	S6
Noise Characteristics	Babble, F16, R1,V1	Car, Factory, R2, V2	Babble, Car, R2, V2	F16, Factory, R1, V1	Car, F16, R1, V1	Babble, Factory, R2, V2

Table 3.2: Verification Results on the degraded Fisher speech dataset.

Exp. #	Training set / Testing set	MFCC / LPC / MFCC-LPC							
		TMR@FMR=10%				minDCF($P_{tar} = 0.01$)			
		1D-Triplet-CNN	UBM-GMM	iVector-PLDA	xVector-PLDA	1D-Triplet-CNN	UBM-GMM	iVector-PLDA	xVector-PLDA
7	F1 / F1	74 / 73 / **85**	54 / 42 / 55	67 / 18 / 70	57 / 63 / 72	7.69 / 7.19 / **5.67**	9.95 / 9.86 / 9.95	8.33 / 9.97 / 7.8	8.53 / 8.8 / 8
8	F1 / F2	55 / 46 / **74**	39 / 40 / 43	46 / 18 / 53	25 / 42 / 54	9.31 / 9.82 / **7.4**	9.92 / 9.81 / 9.9	9.55 / 9.99 / 9.59	9.94 / 9.7 / 9.37
9	F2 / F2	77 / 76 / **84**	56 / 42 / 57	68 / 20 / 72	56 / 56 / 73	6.96 / 7.37 / **5.53**	9.52 / 9.45 / 9.48	8.19 / 9.96 / 8.07	9.25 / 8.96 / 7.62
10	F2 / F1	39 / 36 / **62**	29 / 42 / 32	38 / 22 / 41	29 / 28 / 44	9.84 / 9.88 / **8.66**	9.95 / 9.89 / 9.95	9.54 / 9.96 / 9.59	9.89 / 9.91 / 9.74

Exp. #	Training set / Testing set	MFCC / LPC / MFCC-LPC			
		Equal Error Rate (EER, in %)			
		1D-Triplet-CNN	UBM-GMM	iVector-PLDA	xVector-PLDA
7	F1 / F1	16 / 17 / **12**	24 / 27 / 23	18 / 43 / 18	22 / 20 / 17
8	F1 / F2	23 / 25 / **17**	29 / 31 / 26	26 / 44 / 24	37 / 30 / 23
9	F2 / F2	16 / 16 / **13**	25 / 31 / 24	19 / 41 / 17	24 / 23 / 16
10	F2 / F1	30 / 31 / **22**	29 / 28 / 27	31 / 41 / 29	37 / 31 / 27

Data Subset	F1	F2
Noise Characteristics	Babble, R1,V1	F16, R1, V1

noise profiles. Therefore, normalizing the data with respect to a single type of noise, present in the training set, does not generalize across the validation and testing sets degraded with unknown noise profiles. We solve this problem by replacing spectral substraction with input-normalization for every activation layer in our proposed CNN. This ensures a uniform normalization of the data being processed in the CNN at every layer, independent of the type of noise added to the input data. For this purpose, we compared the performance of 'Batch Normalization with ReLU activation' to SELU [92] activation, a recently proposed alternative to the former approach, and found significant performance benefits over the former. We have also used the 'alpha dropout layer', as suggested in [92], for maintaining the self-normalizing property of the SELU activations. Therefore, SELU activation layer coupled with alpha dropout helps in improving the generalizablity, as seen in section 4.4, of our proposed CNN architecture across different types of audio degradations.

Table 3.3: Verification Results on the original and degraded, NIST SRE 2008 and 2010 datasets.

Exp. #	Training set / Testing set	MFCC / LPC / MFCC-LPC							
		TMR@FMR=10%				minDCF($P_{tar} = 0.01$)			
		1D-Triplet-CNN	UBM-GMM	iVector-PLDA	xVector-PLDA	1D-Triplet-CNN	UBM-GMM	iVector-PLDA	xVector-PLDA
11	P1 / P1	89 / 86 / **93**	51 / 47 / 62	85 / 78 / 88	78 / 76 / 85	5.45 / 5.68 / **4.72**	9.14 / 9.87 / 9.59	5.68 / 7.45 / 5.84	8 / 8.18 / 7.19
12	P1 / P2	21 / 18 / **25**	15 / 11 / 11	14 / 17 / 10	19 / 15 / 17	9.95 / 9.98 / 9.96	**9.9** / 9.99 / 9.99	9.91 / 9.94 / 9.98	9.97 / 9.94 / 9.95
13	P3 / P3	84 / 80 / **89**	58 / 44 / 17	82 / 72 / 41	75 / 65 / 72	6.39 / 6.89 / **5.36**	8.83 / 9.67 / 9.99	6.36 / 7.51 / 9.58	8.35 / 8.76 / 8.25
14	P4 / P4	75 / 73 / **84**	44 / 34 / 11	60 / 28 / 22	58 / 54 / 66	7.24 / 7.77 / **6.62**	9.15 / 9.85 / 9.99	8.5 / 9.84 / 9.97	9 / 9.16 / 8.58
15	P3 / P4	49 / 47 / **56**	43 / 34 / 15	28 / 20 / 15	31 / 35 / 52	9.4 / 9.35 / **9**	9.7 / 9.87 / 10	9.93 / 9.94 / 9.98	9.72 / 9.92 / 9.28
16	P4 / P3	37 / 27 / **56**	31 / 28 / 14	51 / 21 / 20	45 / 52 / 47	9.57 / 9.98 / **9.01**	9.99 / 9.99 / 9.99	9.23 / 9.95 / 9.99	9.55 / 9.65 / 9.5

Exp. #	Training set / Testing set	MFCC / LPC / MFCC-LPC			
		Equal Error Rate (EER), in %			
		1D-Triplet-CNN	UBM-GMM	iVector-PLDA	xVector-PLDA
11	P1 / P1	10 / 11 / **8**	29 / 26 / 23	12 / 15 / 10	14 / 15 / 11
12	P1 / P2	45 / 44 / **39**	44 / 44 / 49	45 / 44 / 47	43 / 46 / 45
13	P3 / P3	12 / 14 / **10**	24 / 28 / 40	13 / 17 / 28	15 / 19 / 16
14	P4 / P4	16 / 17 / **12**	31 / 37 / 46	22 / 36 / 37	20 / 22 / 19
15	P3 / P4	26 / 28 / 23	31 / 34 / 42	35 / 41 / 43	34 / 31 / **22**
16	P4 / P3	32 / 37 / **23**	36 / 37 / 44	24 / 40 / 41	27 / 22 / 27

Data Subset	P1	P2	P3	P4
Noise Characteristics	NIST SRE 08	NIST SRE 10	NIST SRE 08 + Babble	NIST SRE 08 + F16

Table 3.4: Verification Results under varying audio length on the NIST SRE 2008 dataset

Length of Audio (in seconds)	MFCC / LPC / MFCC-LPC							
	TMR@FMR=10%				minDCF($P_{tar} = 0.01$))			
	1D-Triplet-CNN	UBM-GMM	iVector-PLDA	xVector-PLDA	1D-Triplet-CNN	UBM-GMM	iVector-PLDA	xVector-PLDA
3.5	90 / 88 / **94**	53 / 46 / 61	78 / 75 / 86	78 / 74 / 81	4.98 / 5.25 / **4.3**	8.95 / 9.91 / 9.76	6.25 / 7.8 / 6.05	7.61 / 8.31 / 7.41
3	90 / 88 / **94**	52 / 45 / 60	77 / 71 / 84	76 / 71 / 79	5.06 / 5.39 / **4.26**	9.05 / 9.95 / 9.8	6.62 / 7.94 / 6.43	8.19 / 8.74 / 7.74
2.5	89 / 88 / **94**	50 / 43 / 58	70 / 67 / 82	69 / 66 / 75	5.25 / 7.07 / **4.15**	9.15 / 9.94 / 9.79	6.94 / 8.34 / 6.75	8.61 / 8.9 / 8.17
2	87 / 85 / **93**	48 / 43 / 55	66 / 59 / 78	61 / 57 / 66	5.17 / 5.71 / **4.51**	9.59 / 9.95 / 9.79	7.66 / 8.8 / 7.25	9.08 / 9.48 / 8.84
1.5	86 / 84 / **91**	44 / 40 / 50	58 / 48 / 68	52 / 47 / 57	6.11 / 5.89 / **4.84**	9.79 / 9.95 / 9.86	8.9 / 9.49 / 8.41	9.28 / 9.73 / 9.16
1	80 / 79 / **87**	38 / 34 / 46	40 / 33 / 54	37 / 34 / 41	6.98 / 6.95 / **5.76**	9.87 / 9.97 / 9.84	9.4 / 9.81 / 9.25	9.72 / 9.96 / 9.65
0.5	65 / 63 / **76**	27 / 26 / 32	22 / 19 / 31	19 / 20 / 20	8.44 / 8.41 / **7.3**	9.9 / 9.97 / 9.9	9.95 / 9.94 / 9.82	9.92 / 9.96 / 9.89

Length of Audio(in seconds)	MFCC / LPC / MFCC-LPC			
	Equal Error Rate (EER, in %)			
	1D-Triplet-CNN	UBM-GMM	iVector-PLDA	xVector-PLDA
3.5	9 / 10 / **7**	28 / 27 / 22	14 / 16 / 11	14 / 15 / 12
3	9 / 10 / **7**	29 / 26 / 23	16 / 17 / 12	15 / 17 / 14
2.5	10 / 11 / **7**	30 / 27 / 24	17 / 19 / 13	17 / 18 / 15
2	10 / 12 / **8**	31 / 28 / 25	20 / 22 / 15	20 / 21 / 18
1.5	11 / 12 / **9**	33 / 31 / 28	24 / 27 / 18	24 / 26 / 22
1	13 / 14 / **11**	36 / 33 / 32	30 / 33 / 24	31 / 32 / 29
0.5	20 / 20 / **15**	43 / 39 / 38	41 / 44 / 35	43 / 42 / 40

Table 3.5: Verification Results on degraded TIMIT dataset for comparing the performance of 1D-Dilated CNN architecture with alternate 1D CNN and 2D CNN architectures.

Method	Performance on Testing Set		Performance on Training Set		Number of Model Parameters
	TMR@FMR=10%	minDCF($P_{tar} = 0.01$)	TMR@FMR=10%	minDCF($P_{tar} = 0.01$)	
1D-CNN (with dilation, along feature dimension)	**75.79**	**8.97**	**98.57**	**3.95**	89,696
1D-CNN (with dilation, along time dimension)	19.04	9.64	40.17	9.89	89,696
1D-CNN (with pooling, along feature dimension)	61.3	9.33	88.54	6.95	89,696
2D-CNN (with pooling)	47.02	9.52	93.8	6.17	768,800

3.3.0.5 Cosine Triplet Embedding Loss

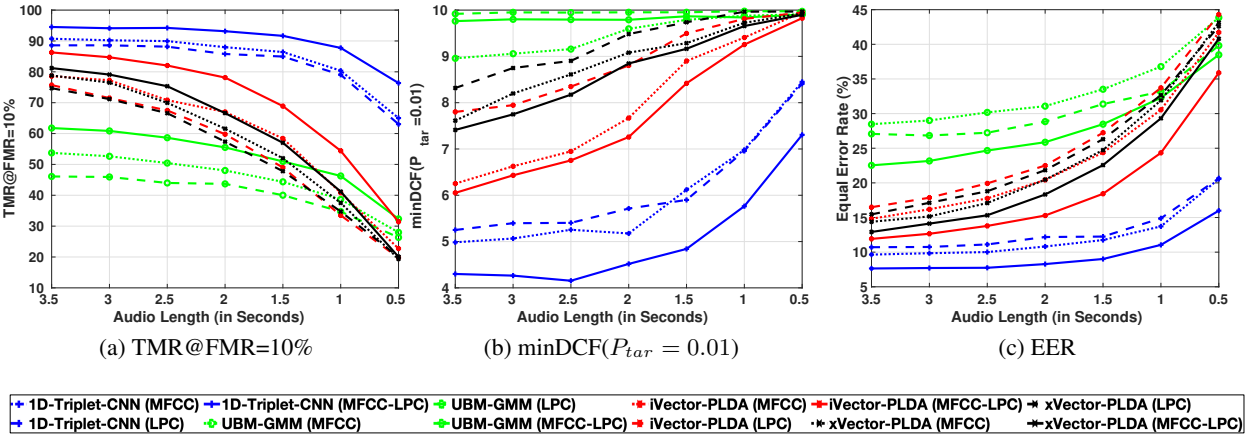
The main aim of triplet based CNNs, as introduced in [147], is to learn an embedding $f(x) \in \Re^d$. Where x is a data sample and $f(x)$ is its embedding in a d-dimensional euclidean space. The embedding is so learnt such that data samples belonging to same class are embedded closer to each other in the d-dimensional space while embedding of samples from different classes are pushed farther apart. Similar to the work in [64], we use cosine similarity metric for learning the

embeddings as it provides for better learning dynamics over euclidean metric in our case.

The cosine triplet embedding loss used for training the 1D-Triplet-CNN model is given by:

$$L(S_a, S_p, S_n) = \sum_{a,p,n}^{N} \cos(f(S_a, S_n)) - \cos(f(S_a, S_p)) + \alpha_{margin} \quad (3.3.1)$$

Here, $L()$ is the cosine triplet embedding loss function. S_a, the anchor sample, and S_p, the positive sample, are speech samples from a subject 'A'. S_n, the negative sample, is a speech sample from another subject 'B', such that $A \neq B$. α_{margin} is the margin of minimum distance between positive and negative samples and is a user tunable hyper-parameter.



(a) TMR@FMR=10% (b) minDCF($P_{tar} = 0.01$) (c) EER

Figure 3.5: (a) TMR@FMR=10%, (b) minDCF($P_{tar} = 0.01$) and (c) EER under varying audio length on the clean NIST SRE 2008 dataset. 1D-Triplet-CNN(MFCC-LPC) performs the best across varying lengths of test audio.

3.4 Datasets and Experiments

3.4.1 Experiments

Throughout the paper we perform speaker verification experiments on a variety of datasets and protocols. For evaluating our proposed algorithm we use:

1. TIMIT [60] Acoustic-Phonetic Continuous Speech Corpus

2. Fisher English Training Speech Part 1 Speech [44] dataset

3. NIST SRE 2008 [1] dataset

4. NIST SRE 2010 [2] dataset

We further use noise data, as detailed in section 4.3.1, from NOISEX-92 [165] dataset under varying levels (0 to 20 dB) of Signal to Noise Ratio (SNR) and reverberations to degrade the speech data in above listed datasets. This is done to demonstrate the performance of our algorithm under degraded audio conditions.

Each of the datasets and corresponding protocols used in the experiments have been designed for evaluating certain aspects of the speaker verification algorithm. The speaker verification experiments on the degraded TIMIT dataset aim at evaluating the generalizability of the algorithms under a variety of audio perturbations. While the experiments on degraded Fisher dataset aim at evaluating the performance of the algorithms in presence of a large number of speakers, hence testing the modeling capacity of the algorithms. The experiments on the NIST SRE experiments aim at comparing the performance of the algorithms on a multilingual speech dataset containing speech data from varying speech types and conditions. We also perform speaker verification experiment on speech samples of varying audio lengths, as explained in section 4.3.2.6, for studying the effect of variation in the length of test audio samples on the performance of speaker verification algorithms. One important point to note is that throughout all our experiments we work with the assumption that only one subject speaks in any given speech sample and there is no overlapping speech from multiple speakers in any audio in the training or testing sets.

3.4.2 Datasets

3.4.2.1 TIMIT Dataset

The TIMIT dataset provides clean speech recordings of 630 speakers. There are 462 speakers in the training set and 168 speakers in the testing set. The dataset consists of eight major dialects of American English. There are ten sessions of 3 seconds each per speaker in the dataset. The text

spoken by the speakers in the training set and test set are disjoint, making the speaker recognition experiments text-independent.

In our experiments, TIMIT dataset was perturbed [20, 74] with different types (given below) of synthetic noise from the NOISEX-92 [165] noise dataset. We refer to this dataset as 'degraded TIMIT' dataset. The audio degradations were added to the TIMIT dataset in simulated room environments with different acoustic properties and reverberation levels, thereby introducing both additive and convolutive noise into the audio data. The synthetically generated noisy datasets have the following noise characteristics:

1. Noise Type: Following four types of noises were added to the TIMIT dataset:

   a)F-16: Noise generated by engine of F-16 fighter aircraft.

   b)Babble: Noise generated by rapid and continuous background human speech.

   c)Car: Noise generated by engine of a car.

   d)Factory: Noise generated by heavy machinery operating in a factory environment.

2. Signal to Noise Ratio (SNR): The resultant noisy datasets were each generated at three different SNR levels, viz., 20 dB, 10dB and 0dB.

3. Room Size: The noisy dataset were generated in a simulated room environment with two different room sizes (4m and 20m, side length of cube), referred to as R1 and R2 in the protocol.

4. Reverberation: Two different reverberation coefficients were used to introduce additional noise in the data, referred to as V1 and V2 in the protocol.

3.4.2.2 Fisher English Training Speech Part 1 dataset

The Fisher English Training Speech Part 1 Speech dataset contains conversational speech data collected over telephone channels between pairs of over $12,000$ speakers. Conversations pertaining to a subset of $6,991$ speakers from the dataset were used in this work. A random subset of 4500 speakers (out of $6,991$ total speakers) was chosen for the training set and remaining speakers were

reserved for the testing set. Since the speech audios in Fisher dataset contains speech from multi-speaker conversations, speech audio pertaining to each speaker in the conversation is segmented out and processed with voice activity detection to remove empty audio segments. The audio of each speaker was then split into smaller audio snippets of 5-second duration each. We extract 50 audio snippets for each speaker. We have also perturbed the Fisher dataset with the F-16 and Babble noise from the NOISEX-92 [165] noise dataset, at a resultant SNR of 10dB. The noisy datasets were generated in a simulated room environment of fixed size (4m, side length of cube), referred to as R1 in the protocol. Fixed amount of reverberation was also used to introduce additional noise in the data, referred to as V1 in the protocol.

3.4.2.3 NIST SRE 2008 and 2010 datasets

National Institute of Standards and Technology (NIST) periodically conducts the NIST Speaker Recognition Evaluation (SRE) challenges to evaluate performance of speaker recognition algorithms under various audio characteristics. In our work, we use the NIST SRE 2008 dataset to train our models. For evaluation we use both the NIST SRE 2008 and 2010 datasets. For our experiments on the NIST SRE 2008 [1] dataset, we use multilingual speech data from 'phonecall' and 'interview' speech types, collected across varied audio conditions labeled as '10-sec', 'long' and 'short2'. We choose a random subset of 1136 speakers for training our algorithms and rest 200 speakers are reserved for evaluation purposes. For cross-dataset speaker verification performance evaluation, we use speech data from all the speakers in the evaluation test set of the NIST SRE 2010 [2] dataset. We have also perturbed the NIST SRE 2008 dataset with synthetic noise from the NOISEX-92 [165] noise dataset. We added F-16 and Babble noise to the NIST SRE 2008 dataset. The Signal to Noise ratio of the resultant *degraded* NIST SRE 2008 dataset is maintained at 0dB.

3.4.3 Features

All experiments using the proposed 1D-Triplet-CNN algorithm and the three baselines of UBM-GMM, iVector-PLDA and xVector-PLDA, as detailed in Section 3.4.4, are done using the MFCC

and LPC feature sets individually. The same experiments have then been repeated for all the algorithms using the fusion of MFCC and LPC feature sets, referred to as MFCC-LPC. This was done to better understand the ability of the different algorithms at combining information from two seemingly complementary feature sets and leveraging it for performing speaker recognition. For the 1D-Triplet-CNN algorithm, as also discussed in Sections 3.3.0.1 and 3.3.0.2, the MFCC and LPC features were fused together into a two channel feature matrix yielding an input feature dimensionality of $40 \times 200 \times 2$. However, for the UBM-GMM, iVector-PLDA and xVector-PLDA algorithms, the MFCC and LPC features (in that order) were concatenated end-to-end at frame level, yielding an input feature dimensionality of 80×200. This was done because the VOICEBOX toolkit's implementation of UBM-GMM and iVector-PLDA algorithms does not support multi-channel feature input.

3.4.4 Experimental Protocols

In all the experiments, across all the datasets, we ensure disjoint speakers in training and testing sets. The split of noise characteristics, however, in the training and testing sets are experimented with both disjoint noise and same noise scenarios, as given in Tables 3.1, 4.2, 4.3. For example, in experiment 1, the training set consists of audio samples that are simulated to be recorded in a room of size R1 and reverberation coefficient V1, with additive background noise of type "Babble" and "F16".

3.4.4.1 UBM-GMM [137] based Speaker Verification Experiments

To obtain baseline performance on the experiments laid out in Tables 3.1, 4.2, 4.3 and 4.5, we train a Universal Background Model (UBM) [137] using data from the speakers in the training set. We evaluate the trained model on verification audio pairs from speakers in the testing set. For evaluation, we adapt the trained UBM to each of the audio samples in a verification pair to obtain two separate speaker-adapted GMM models. Which are then scored against each other to render a match score.

3.4.4.2 iVector-PLDA [63] based Speaker Verification Experiments

To obtain a second baseline performance on the experiments laid out in Tables 3.1, 4.2, 4.3 and 4.5, we perform iVector-PLDA based speaker recognition experiments using the implementation in the MSR identity toolkit [143]. Similar to the protocol for the UBM-GMM experiment, we first train an UBM on audio data from speakers in the training set. The trained UBM is then used to learn a total variability subspace of 400 dimensions, from background statistics. Development i-vectors are then extracted from speech features using the trained total variability subspace and UBM. Finally a Gaussian PLDA model with development i-vectors is learnt. For evaluation, i-vectors are generated for both the audio samples in a verification pair and then they are compared using the trained PLDA model to render a match score.

3.4.4.3 xVector-PLDA [154] based Speaker Verification Experiments

To obtain a neural network based baseline performance for the experiments reported in Tables 3.1, 4.2, 4.3 and 4.5, we use xVector-PLDA. Since the xVector implementation in Kaldi [130] toolkit only supports 24-dimensional MFCC features, we re-implemented the xVector algorithm in PyTorch for enabling support for 40-dimensional MFCC and LPC features and the 80-dimensional MFCC-LPC features. The PyTorch implementation of xVector algorithm was used together with the gaussian PLDA implementation given in the MSR identity toolkit [143] for performing the xVector-PLDA based speaker recognition experiments.

3.4.4.4 1D-Triplet-CNN based Speaker Verification Experiments

The experiments, given in Tables 3.1, 4.2, 4.3 and 4.5, were conducted using the proposed 1D-Triplet-CNN based Speaker Verification algorithm. For training the 1D-Triplet-CNN, we generate data triplets using audio data from speakers in the training set. For evaluation we generate genuine-impostor verification pairs from speakers in the testing set and match them as explained in Section 3.3.

3.4.4.5 Speaker verification experiments on audio samples of varying length

We also perform speaker verification experiments using the UBM-GMM, iVector-PLDA, xVector-PLDA and 1D-Triplet-CNN algorithms on speech data of varying lengths from the NIST SRE 2008 dataset. This experiment is aimed at evaluating the effect of variation in length of test audio on the performance of speaker verification algorithms. In practical scenarios, the probe audio sample is often of limited length in which the amount of usable speech audio is further reduced greatly by audio perturbations. Therefore it is important for speaker verification algorithms to be robust across different lengths of audio samples.

We compare the True Match Rate at a False Match Rate of 10% (TMR@FMR=10%), minimum Detection Cost Function at a priori probability of the specified target speaker, P_{tar}, of 0.01 (minDCF($P_{tar} = 0.01$)) and Equal Error Rate (EER, in $\%$) for both baseline and proposed algorithms on audio samples of varying number of frames. We vary the audio length from 3.5 to 0.5 seconds in steps of 0.5 second.

3.4.4.6 Speaker Verification Experiments for comparing the performance benefits of *dilated* 1D convolutions over traditional 1D and 2D CNN architectures

We also perform additional speaker recognition experiments, as given in Table 3.5, for experimentally validating the design choice of using dilated 1D convolutions along the feature dimension. We use data from degraded TIMIT dataset for these experiments. Audio data degraded with Babble and F16 noise is used to train the models, while data in the testing set is perturbed with Car and Factory noise. All the CNN designs are kept mutually identical in all other aspects (e.g. number of layers, number of input and output channels etc.), except the shape of the convolution filters used and the usage of dilation versus pooling operation. Following CNN designs are explored in these set of experiments:

- 1D-CNN (with dilation, along feature dimension): This is the design introduced in our proposed method where dilated 1D convolution filters are learnt along the feature dimension for extracting speaker dependent information at frame-level.

• 1D-CNN (with dilation, along time dimension): This design uses 1D dilated convolution filters learnt along the time dimension for extracting speaker dependent information at multiple temporal scales.

• 1D-CNN (with pooling, along feature dimension): This design replaces the use of dilated 1D convolution layers as done in the proposed 1D-Triplet-CNN architecture with regular 1D convolution layers paired with average pooling layers.

• 2D-CNN (with pooling): This design replaces the use of dilated 1D convolution layers as done in the proposed 1D-Triplet-CNN architecture with regular 2D convolution layers paired with average pooling layers.

We report performance of the different CNN designs on both the training and testing sets along with their number of learnable model parameters.

3.4.4.7 Score-level fusion experiments for combining speaker recognition models trained on MFCC and LPC features separately

We also performed score-level fusion of the speaker recognition models trained individually on MFCC and LPC features, using the sum and product fusion rules. We performed these experiments on the six subsets (S1 - S6) of the degraded TIMIT dataset, defined in Table 3.1, to compare the performance benefits of the score-level and feature-level fusion approaches. For all the baseline and proposed algorithms, both sum and product rule based score-level fusion approaches outperformed the models trained individually on the LPC features. However, they failed to outperform the models trained on the MFCC features alone. When compared to the performance of feature-level fusion approach, the score-level fusion approaches lag behind by $\sim 22\%$ from the 1D-Triplet-CNN algorithm and by $\sim 2\%$ from the xVector-PLDA algorithm. However, for the UBM-GMM and iVector-PLDA algorithms, the score-level fusion approaches outperform the feature-level fusion approach by $\sim 10\%$ and $\sim 2\%$, respectively. Since the score level fusion strategies did not appear

to benefit the overall speaker verification performance for the 1D-Triplet-CNN and xVector-PLDA algorithms, they have been excluded from further consideration.

3.5 Results and Analysis

The results of the verification experiments are presented in Tables 3.1, 4.2, 4.3 and 4.5. We have chosen to report True Match Rate at False Match Rate of 10% (TMR@FMR=10%), minimum Detection Cost Function at a priori probability of the specified target speaker, P_{tar}, of 0.01 (minDCF($P_{tar} = 0.01$)) and Equal Error Rate (EER, in $\%$) as our performance metric for comparison of the baseline methods and the proposed method. The Detection Error Tradeoff (DET) curves are given in Figures 3.4 and 4.6. Additionally, we also determined and reported the subsets of test data pairs that were correctly matched using the proposed and the baseline algorithms at False Match Rate of 10%. This was used to determine the proportion of the test data pairs where the proposed algorithm performed better or worse than the baseline algorithms.

• The proposed algorithm vastly outperforms the baseline algorithms, in majority of the experiments given in Tables 3.1, 4.2, 4.3 and 4.5, when trained/tested on MFCC and LPC features separately and also when fused together. Also, it is interesting to note that, unlike the baseline algorithms, the proposed algorithm successfully fuses the MFCC and LPC features to gain consistent performance benefits over the individual features, across all the experiments. The main reason for the performance improvement can be attributed to the design of the 1D-Triplet-CNN architecture, which: (a) successfully extracts speaker dependent features from MFCC and LPC features drawn from degraded audio signals and (b) successfully combines the extracted speaker dependent features to learn a highly discriminative joint-embedding for improving speaker recognition performance.

• On average, across the six experiments in Table 3.1, UBM- GMM, iVector-PLDA, xVector-PLDA, and 1D-Triplet-CNN correctly verified the same 34.97% of the test samples. 1D-Triplet-CNN correctly verifies an additional 14.08% of the test samples over xVector-PLDA, 28.72% over iVector-PLDA, and 33.48% over UBM-GMM based algorithms. However, the 1D-Triplet-CNN

based algorithm fails to correctly verify 4.4% of the test samples that were correctly verified by all the baseline algorithms.

- On average, across the six experiments in Table 3.1, the best baseline performance (TMR at FMR=10%) is achieved by xVector-PLDA(MFCC-LPC) algorithm. The proposed 1D-Triplet-CNN(MFCC-LPC) algorithm further improves upon the best average baseline performance, by 12%. It also improved the average EER from 18% to 15% and minDCF($P_{tar} = 0.01$) from 8.46 to 7.89.

- On average, across the four experiments in Table 4.2, UBM- GMM, iVector-PLDA, xVector-PLDA, and 1D-Triplet-CNN correctly verified the same 47.5% of the test samples. 1D-Triplet-CNN correctly verifies an additional 17.18% of the test samples over xVector-PLDA, 13.55% over iVector-PLDA, and 16.06% over UBM-GMM based algorithms. However, the 1D-Triplet-CNN based algorithm fails to correctly verify 4.3% of the test samples that were correctly verified by all the baseline algorithms.

- On average, across the four experiments in Table 4.2, the best baseline performance (TMR at FMR=10%) is achieved by xVector-PLDA(MFCC-LPC) algorithm. The proposed 1D-Triplet-CNN(MFCC-LPC) algorithm further improves upon the best average baseline performance, by almost 16%. It also improved the average EER from 20% to 16% and minDCF($P_{tar} = 0.01$) from 8.68 to 6.81.

- On average, across the six experiments in Table 4.3, UBM- GMM, iVector-PLDA, xVector-PLDA, and 1D-Triplet-CNN correctly verified the same 36.35% of the test samples. 1D-Triplet-CNN correctly verifies an additional 17.76% of the test samples over xVector-PLDA, 20.62% over iVector-PLDA, and 25.33% over UBM-GMM based algorithms. However, the 1D-Triplet-CNN based algorithm fails to correctly verify 6.73% of the test samples that were correctly verified by all the baseline algorithms.

- On average, across the six experiments in Table 4.3, the best baseline performance (TMR at

FMR=10%) is achieved by xVector-PLDA(MFCC) algorithm. The proposed 1D-Triplet-CNN(MFCC-LPC) algorithm further improves upon the best average baseline performance, by almost 11%. It also improved the average EER from 23% to 19% and minDCF($P_{tar} = 0.01$) from 8.79 to 7.44.

• In the experimental results given in Table 4.5 and illustrated in Figure 4.6, we notice a decreasing trend in verification performance, i.e., decrease in TMR at FMR=10% and increase in minDCF($P_{tar} = 0.01$) and EER, with decrease in length of audio samples in the testing data, across all the algorithms. However, it is interesting to note the vastly different rates of decrease in performance across all the algorithms. The iVector-PLDA and xVector-PLDA baseline algorithms exhibit a comparatively sharper decrease in performance when compared to others. On average, the iVector-PLDA and xVector-PLDA algorithms, on MFCC-LPC feature set, lose 55% and 60% performance (TMR@FMR=10%), respectively, when the audio length decreases from 3.5 to 0.5 seconds. The UBM-GMM and 1D-Triplet-CNN, however, only lose about 30% and 18% performance (TMR@FMR=10%), respectively in the same experimental setting.

• The aggregate TMR@FMR=10% for 1D-Triplet-CNN, on MFCC-LPC feature set given in Table 4.5, is vastly superior at $90 \pm 6\%$ as compared to $52 \pm 10\%$ for UBM-GMM, $69 \pm 20\%$ for iVector-PLDA, and $60 \pm 22\%$ for xVector-PLDA. The 1D-Triplet-CNN, on MFCC-LPC feature set, maintains an aggregate minDCF($P_{tar} = 0.01$) of $5.01 \pm 1.14\%$ as compared to $9.82 \pm 0.04\%$ for UBM-GMM, $7.70 \pm 1.46\%$ for iVector-PLDA, and $8.69 \pm 0.95\%$ for xVector-PLDA. The 1D-Triplet-CNN, on MFCC-LPC feature set, also maintains an EER of $9.62 \pm 3.04\%$ as compared to $27.91 \pm 5.73\%$ for UBM-GMM, $18.89 \pm 8.61\%$ for iVector-PLDA, and $21.90 \pm 10.08\%$ for xVector-PLDA.

• Thus, we can establish that the performance of the 1D-Triplet-CNN is relatively robust to the variation in length of audio samples in the testing data. This can be attributed to the architecture of 1D-Triplet-CNN that performs dilated 1D convolutions only along individual frames and is independent of the length of context. However, the iVector-PLDA and xVector-PLDA algorithms use statistic pooling across the frames in an audio sample for characterizing it. Reliability of such

statistic pooling operations are heavily dependent on the number of available audio frames. Thus, reducing the length of audio has a greater detrimental effect on iVector-PLDA and xVector-PLDA algorithms.

• Across the four experiments, given in Table 3.5, the best performance is achieved by our proposed architecture design of using dilated 1D convolutions along the feature dimension. We compare the effect of performing dilated 1D convolutions along the feature and time axes individually and found the former to perform vastly superior. This validates the assumption of speaker-dependent voice characteristics to be stable only within individual frames of short time scales (25ms). We further compare the effect of using dilation against pooling operation with 1D convolutions (along the frames) for increasing the receptive field of 1D-convolution filters deeper in the network. As evidenced in the results, pooling operation results in inferior performance as compared to dilation operation, thereby confirming the detrimental effect of data loss incurred by the pooling operation. This supports the design choice of using dilation over pooling operation. Finally, we also train a network with 2D convolution filters paired with average pooling operation, that is popularly used in image classification networks. It is interesting to note that the second-best training performance is attained by this design, while falling behind considerably on testing performance when compared to architectures using 1D convolution filters (along frames). This indicates signs of overfitting in case of 2D-CNN. It is also important to note that the proposed 1D-CNN architecture has only 89K trainable model parameters compared to the 768K model parameters on 2D-CNN and 4.2M parameters in the xVector network design. The 1D-Triplet-CNN model is therefore much easier to train and converge using limited data and computational resources.

3.6 Implementation and Reproducibility

The 1D-Triplet-CNN model was implemented using PyTorch [128] toolkit and trained using the Adam optimizer [88] with a starting learning rate of 0.001 for 150 epochs. The α_{margin} hyper-parameter at the value of 0.25, in our cosine triplet embedding loss, was found to provide the best trade-off between time-to-convergence and generalizability. A higher value of α_{margin}

increased the time-to-convergence considerably while only improving the performance marginally. On the other hand, reducing the value of α_{margin} below 0.25 led to a loss of generalizability as the network failed to separate harder negative samples from positive samples.

3.7 Conclusion

Noise in audio data often distorts the speaker dependent characteristics present in it, thereby confounding speaker verification methods. MFCC as a speech representation technique is not very robust to audio degradations [38, 66]; therefore, speaker recognition performance of methods that solely rely on MFCC features will suffer in the presence of audio degradations. In contrast, the 1D-Triplet-CNN algorithm, that combines MFCC with LPC in a systematic manner, is observed to be robust to a wide range of audio degradations as evidenced in the experimental results. When compared to xVector-PLDA, the 1D-Triplet-CNN algorithm using MFCC-LPC features, improves the average TMR by 12% on the degraded-TIMIT dataset, 16% on the degraded-Fisher dataset and 11% on the degraded NIST SRE 2008 and 2010 datasets at FMR=10%.

In this chapter, we developed a method to strategically combine two complimentary feature sets—MFCC and LPC—for improving speaker recognition performance in degraded audio signals. However, the underlying MFCC and LPC features are hand-crafted and do not adapt well across all the scenarios. As shown in our experiments, while the proposed method outperforms all the baseline methods by a substantial margin, it still fails to correctly verify almost 14% of the samples in the degraded-TIMIT dataset, 16% of the samples in the degraded-Fisher dataset and almost 21% of the samples in the clean and degraded NIST SRE 2008 and 2010 datasets. Therefore, in the next chapter we introduce a method for extracting speaker dependent speech features directly from raw audio data, thus removing our reliance on MFCC and LPC based speech features.

CHAPTER 4

DISCOVERING FEATURES FROM RAW AUDIO FOR SPEAKER RECOGNITION IN DEGRADED AUDIO SIGNALS

Portions of this chapter appeared in the following publication:

Chowdhury, Anurag, and Arun Ross. "Discovering Features from Raw Audio for Speaker Recognition in Degraded Audio Signals." IEEE Transactions on Pattern Analysis and Machine Intelligence (2021- To be submitted).

4.1 Introduction

In the previous chapters, we focused on developing 1D-CNN based techniques for extracting speaker dependent speech characteristics from MFCC and LPC features and fusing them for performing noise-robust speaker recognition in degraded audio signals. In this chapter, we will introduce a method, called DeepVOX, for automatically discovering features directly from raw speech audio for performing speaker recognition in degraded audio signals.

Automatic speaker recognition algorithms typically use pre-defined filterbanks, such as Mel-Frequency and Gammatone filterbanks, for characterizing speech audio. The design of these filterbanks is based on domain-knowledge and limited empirical observations. The resultant features, therefore, may not generalize well to different types of audio degradation. In this work, we propose a deep learning-based technique to induce the filterbank design from vast amounts of speech audio. The purpose of such a filterbank is to extract features robust to non-ideal audio conditions, such as degraded, short duration, and multi-lingual speech. To this effect, a 1D convolutional neural network is designed to learn a time-domain filterbank called DeepVOX directly from raw speech audio. Secondly, an adaptive triplet mining technique is developed to efficiently mine the data samples best suited to train the filterbank. Thirdly, a detailed ablation study of the DeepVOX filterbanks reveals the presence of both vocal source and vocal tract characteristics in the extracted

features. Experimental results on VOXCeleb2, NIST SRE 2008, 2010 and 2018, and Fisher speech datasets demonstrate the efficacy of the DeepVOX features across a variety of audio degradations, multi-lingual speech data, and varying-duration speech audio. The DeepVOX features also improve the performance of existing speaker recognition algorithms, such as the xVector-PLDA and the iVector-PLDA.

Automatic speaker recognition aims at recognizing an individual from their voice in speech audio. The performance of a speaker recognition algorithm relies on its ability to extract speaker-dependent characteristics from the speech audio. It is, therefore, important to design robust feature extraction algorithms that can efficiently characterize a speaker's voice. Speaker-dependent features can be captured at multiple levels of abstraction, such as short-term spectral features and prosodic features (among others). Each type of speech feature is suited for modeling a fixed set of speech characteristics. Hence, depending upon the scenario, the choice of speech feature largely impacts the speaker recognition performance.

Short-term spectral features, for example, are extracted from short speech segments to efficiently model the vocal tract of the speaker. While such features are effective in clean speech scenarios, they are not robust to audio degradations [66]. Prosodic features, on the other hand, are derived from longer speech segments like syllables, words, and utterances to efficiently capture the speaking style of a speaker [102]. While prosodic features are known to be relatively robust to audio degradations, they typically underperform the short-term spectral features in low-noise scenarios [102]. Therefore, the choice between different types of speech features can be based on the application scenario.

Audio degradations, such as background noise, are one of the most common and challenging scenarios for speaker recognition. While knowledge of the type and extent of audio degradation may help mitigate its negative effects to a certain extent, but, noise estimation in speech audio in itself is a challenging task. For example, speech audio recorded in a coffee shop might suffer from various types of background noise like babble noise from customers and machinery noise from coffee machines. The amount of such audio degradations depends on the number of people or

machinery used in the background at the time of audio recording. Therefore, it is important to learn robust speech features that can adapt to the noise present in data and provide better generalizability without any prior knowledge about the type and amount of audio degradations.

A majority of the latest deep learning-based speaker recognition techniques such as xVector-PLDA [154], 1D-Triplet-CNN [42], and VGGVox [43] rely on handcrafted speech features such as Mel-spectrograms,Mel-frequency Cepstral Coefficients (MFCC), and Linear Predictive Coding (LPC) for performing speaker recognition. However, the representation capability of such features is known to vary with the quality of input audio [66], thus affecting the corresponding speaker recognition performance. This is where we position our work with the currently existing literature. We propose a Convolutional Neural Network (CNN) based approach for learning a filterbank, referred to as DeepVOX, directly from raw speech audio and extracting robust speaker-dependent speech features. The proposed DeepVOX features are then combined with 1D-Triplet-CNN [42], a CNN based speech feature embedding technique, to perform speaker verification. We further propose an adaptive triplet mining technique to improve the performance of triplet learning-based models such as the 1D-Triplet-CNN.

In our experiments, we also demonstrate the compatibility and the associated performance benefits of the DeepVOX features with some of the existing speaker recognition algorithms such as RawNet2 [84], xVector-PLDA [154] and iVector-PLDA [50]. We further study the impact of a large variety of audio degradations, multi-lingual speech data, and varying length speech audio on the representation capability of DeepVOX features. Finally, we also perform a detailed ablation study of the proposed method and conducted a frequency analysis of the learned DeepVOX filterbanks.

In the next section, we discuss and analyze the voice features encoded by some popular speech representation techniques such as MFCC [120] and LPC [101]. We also compare these techniques with the proposed DeepVOX features and discuss their utility in different scenarios.

In this paper, we propose a new approach for extracting noise-robust short-term speech features from raw audio data using 1D-Convolutional Neural Networks (1D-CNN). We draw design choices and inspiration from our previous works on 1D-CNN [38] and 1D-Triplet-CNN [42] based

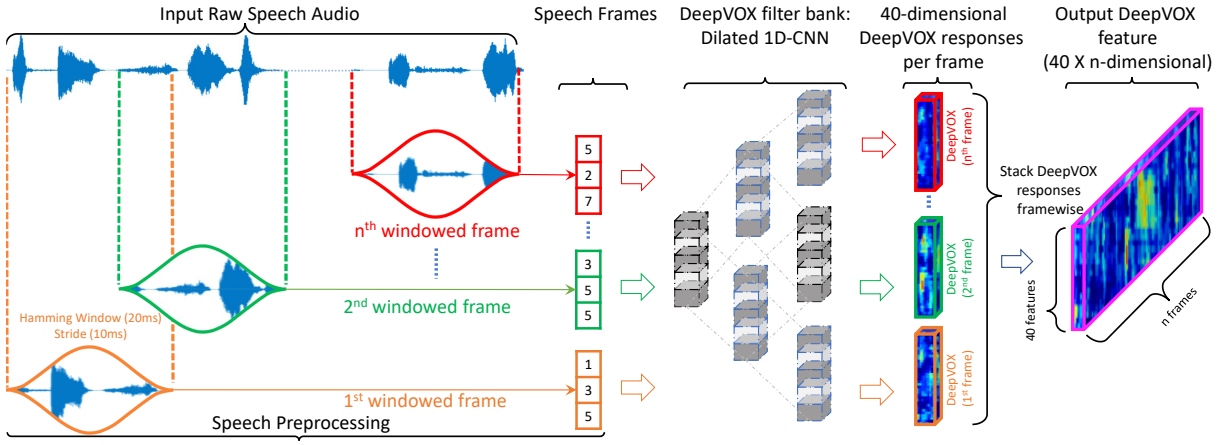


Figure 4.1: A visual representation of the proposed Dilated 1D-CNN based DeepVOX feature extraction process.

architectures for performing speaker identification and verification respectively from degraded audio signals. However, both of these works use MFCC and LPC-based feature representation as input to their network architecture and are thereby limited by the representation power of MFCC and LPC features. We, instead, propose a 1D-CNN [38] based feature extraction module, termed as *DeepVOX*, to learn and extract speech feature representation directly from raw audio data, in time-domain itself. The DeepVOX learns filterbanks directly from a large quantity of degraded raw speech audio samples, thereby laying emphasis on learning highly discriminative speech audio features robust to audio degradations.

Note that, unlike the work in [132], we learn the proposed DeepVOX filterbank without imposing any constraints on the design of the constituent filters. Also, unlike any of the current raw-waveform based speaker recognition methods [118, 119, 132], we demonstrate the compatibility of the proposed DeepVOX features with some latest deep learning-based speaker recognition methods such as xVectors [154] and 1D-Triplet-CNN [42] and even on classical non-deep learning-based methods such as iVector-PLDA [50]. The next few sections present our proposed DeepVOX architecture for feature extraction and discuss its integration in the 1D-Triplet-CNN [42] framework for performing speaker recognition. We also perform an extensive experimental evaluation of the proposed DeepVOX features under a large variety of speech-conditions such as degraded audio, multi-lingual speech, and short duration speech, to demonstrate its performance benefits.

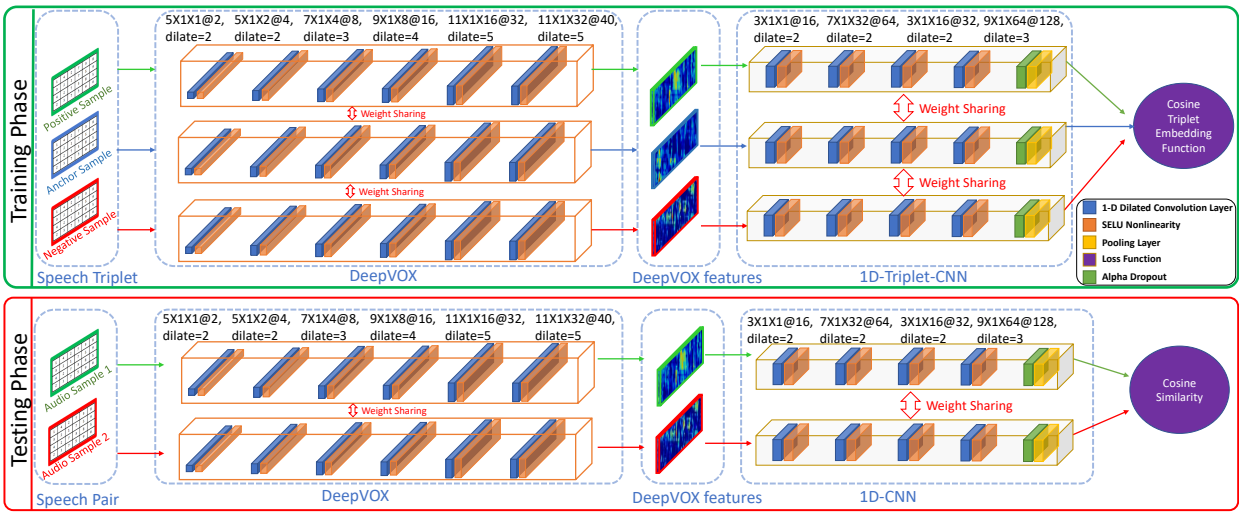


Figure 4.2: A visual representation of the training and testing phases of the proposed DeepVOX architecture. A 1D-Triplet-CNN is used to train the DeepVOX on speech triplets. A siamese 1D-CNN is used to evaluate the trained DeepVOX on pairs of speech audio.

4.2 Proposed Algorithm

In the previous section, we discussed some of the popular speech feature extraction techniques. Depending upon the type of the features being extracted, the algorithms were further categorized into four different feature categories (given in Table 1.1). As discussed, human vocal tract significantly contributes to the majority of speaker dependent features in the human voice. Short-term spectral features are, therefore, well-suited for speaker recognition due to their ability to model the human vocal tract. In the scope of this work, we propose a method for learning a new type of short-term speech features, referred to as *DeepVOX features*, using 1D-Convolutional Neural Networks (1D-CNN). It is important to note that, unlike short-term spectral feature extraction algorithms like MFCC, where the extracted speech features are not specifically geared towards speaker recognition, our proposed algorithm learns to extract features directly from raw speech data, specifically suited for the task of speaker recognition.

4.2.1 Short-term Speech Feature Extraction Using DeepVOX

In this work, we use the proposed DeepVOX feature extractor jointly with a 1D-Triplet-CNN [42]-based feature embedding network for performing speaker recognition. The 1D-Triplet-CNN [42] was initially developed for performing speaker verification in degraded audio signals by combining the MFCC and LPC features into a joint-embedding space. However, here the 1D-Triplet-CNN network is used jointly with the DeepVOX to map the DeepVOX features to a highly discriminative speaker embedding space. The proposed joint architecture (see Figure 7.1), also referred to as 1D-Triplet-CNN(DeepVOX), consists of four separate units described below:

4.2.1.1 Speech Preprocessing

A single channel digital speech audio is usually represented by a one-dimensional vector of real values whose length varies with the time duration and sampling frequency of the audio. We use a Voice Activity Detector [13] to remove non-speech parts of the input audio and restrict the resultant audio to a maximum duration of 2 seconds sampled at a frequency of 8000Hz. This also serves as a data augmentation technique as any audio sample more than 2 seconds long is split into multiple smaller audio samples of length 2 seconds each, thereby increasing the overall number of data samples. The resulting speech audio vector is then *framed and windowed* into multiple smaller audio clips, called *speech units*, using a hamming window of temporal length 20ms and temporal stride of 10ms, as shown in Figure 4.1. Therefore, each speech unit of duration 20ms sampled at 8000Hz is represented by an audio vector of length 160. The running window extracts a *speech unit* every 10ms from a 2sec long input audio, thereby extracting around 200 *speech units* per audio sample. These *speech units* are then stacked horizontally to form a two-dimensional speech audio representation called *speech frame*, each having a physical dimension of 160×200. The extracted speech frames are then made into *speech frame triplets* for inputting into the proposed DeepVOX architecture.

4.2.1.2 Speech Frame Triplets

The authors in [147] introduced the idea of triplet based CNNs. As illustrated in Figure 7.1, our DeepVOX architecture takes a *speech frame triplet* D_t as input. A *speech frame triplet* D_t is defined as a tuple of three speech frames: $D_t = (S_a, S_p, S_n)$ Here, S_a, the anchor sample, and S_p, the positive sample, are two different speech samples from a subject 'X'. S_n, the negative sample, is a speech sample from another subject 'Y', such that $X \neq Y$.

4.2.1.3 DeepVOX

The DeepVOX architecture, as given in Figure 7.1, takes as speech frame triplet as input. DeepVOX processes each speech frame in the triplet to produce a corresponding short term spectral representation, thereby generating a corresponding triplet of *DeepVOX features*. The design of the DeepVOX architecture primarily comprises of 1D Dilated Convolutional Layers [42] and SELU [92] (Scaled Exponential Linear Units) non-linearity. The one dimensional filters are so designed that they only learn features from within *speech units* in a *speech frame* and not across them. This follows the assumption that the speaker dependent characteristics within each speech unit is independent of other speech units in the speech frame. Each 160 dimensional speech unit within a speech frame is processed by layers of 1D Dilated Convolutional Layers to generate 40 filter responses, which constitute the corresponding short-term spectral representation. These 1D Dilated Convolutional Layers interlaced with SELU non-linearity here are designed to jointly represent a filterbank, which unlike the Mel-filterbank or the Gammatone filterbank, is specifically learned for extracting speaker dependent characteristics.

4.2.1.4 1D-Triplet-CNN

The architecture of 1D-Triplet-CNN comprises of interlaced 1D-Dilated-Convolutional layers and SELU non-linearity, followed by alpha dropout and pooling layers. The use of *'dilated convolutions'* over *'convolutions followed by pooling layers'* is motivated by the work done in Wavenet [124],

where the authors use dilated convolutions to increase the receptive field size nonlinearly with a linear increase in number of parameters. In context of 1D-Triplet-CNN, 1D dilated convolutions allow the network to learn sparse relationships between the feature values within a speech unit leading to significant performance benefits. The 1D-Triplet-CNN architecture [42] is designed for learning speaker dependent speech embedding from triplets of *DeepVOX features* generated by the proposed DeepVOX. The three parallel network branches in the 1D-Triplet-CNN architecture learn and share a common set of weights (see Figure 7.1). The aim of the 1D-Triplet-CNN architecture is to transform the *DeepVOX feature* triplet input into a triplet of embeddings, where the intra-class samples are embedded closer to each other and inter-class samples are embedded farther apart. This embedding learning process is ensured by the cosine triplet embedding loss.

4.2.1.5 Cosine Triplet Embedding Loss

The cosine triplet embedding loss [42] is a modification upon the triplet loss intially introduced in [147] by replacing the euclidean distance metric with cosine similarity. As noted in [42], using cosine similarity leads to a faster convergence and more stable learning due to its bounded nature. The triplet loss [147] is designed to learn an embedding $g(f(x)) \in \Re^d$, where $f(x)$ is DeepVOX feature of speech frame x and $g(x)$ is its embedding in a d-dimensional euclidean space ($\Re^d$). In this work, d is set to 128. The embedding is so learned that the intra-class samples are embedded closer to each other than the inter-class samples.

The cosine triplet embedding loss is designed to work on data triplets and its mathematical formulation, as introduced in [42], is given by :

$$L(S_a, S_p, S_n) = \sum_{a,p,n}^{N} \cos(g(f(S_a)), g(f(S_n))) - \cos(g(f(S_a)), g(f(S_p))) + \alpha_{margin} \quad (4.2.1)$$

Here, $L(\cdot, \cdot, \cdot)$ is the cosine triplet embedding loss function. S_a (the anchor sample) and S_p (the positive sample) are two different speech samples from a subject 'X'. S_n (the negative sample) is a speech sample from another subject 'Y', such that $X \neq Y$. α_{margin} is the margin of the minimum distance between positive and negative samples and is a user tunable hyper-parameter.

In the training phase, the task of the loss function, as mentioned in section 4.2.1.4, is to help the network learn the similarity between the anchor sample and the positive sample and the dissimilarity between the anchor sample and the negative sample. As illustrated in Figure 7.1, both the DeepVOX and the 1D-Triplet-CNN networks are trained jointly in our proposed methodology. This has the benefits of simultaneously learning both the embedding space using the 1D-Triplet-CNN and the feature space using the DeepVOX.

In the testing phase (see Figure 7.1) we arrange the trained DeepVOX and 1D-Triplet-CNN networks into a siamese network, i.e. only two identical copies of the trained networks are needed. For testing the network we provide a data pair D_p as input to the CNN, given by:

$$D_p = (S_1, S_2)$$

Here, S_1 and S_2 are speech frames from subjects 'X' and 'Y'. The match score ($Score_{match}$) for the given speech pair is computed using the cosine similarity metric as follows:

$$Score_{match}(S_1, S_2) = \cos(g(f(S_1)), g(f(S_2))) \tag{4.2.2}$$

Here, $g(.)$ is the 1D-Triplet-CNN and $f(.)$ is the DeepVOX network. Under ideal conditions, the match score for a data pair from same subject should be close to 1, while the match score for a data pair from different subjects should be close to -1.

4.2.1.6 Adaptive Triplet Mining for Online Triplet Selection

The effectiveness and generalizability of any network trained using the triplet learning paradigm, such as 1D-Triplet-CNN [42], depends on the difficulty of the training triplets. The authors in [42] trained their proposed 1D-Triplet-CNN algorithm using offline-generated triplets for performing their speaker recognition experiments. However, the effectiveness and computational-feasibility of offline-triplet generation for evenly sampling a speech dataset drastically reduces with the increase in the number of training samples. Online-triplet generation is, therefore, chosen to effectively train the 1D-Triplet-CNN for our experiments. While the majority of online-triplet generation

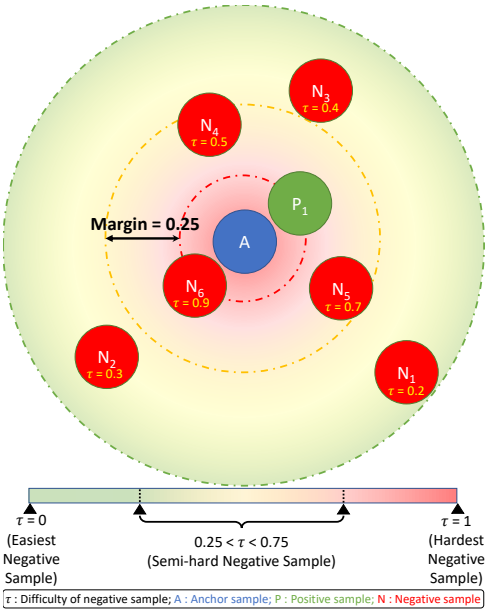


Figure 4.3: A visual representation of adaptive triplet mining used to train the DeepVOX architecture using 1D-Triplet-CNN.

techniques use either hard or semi-hard triplet mining [147], we propose a curriculum learning-based [24] *adaptive triplet mining* technique.

In adaptive triplet mining, at a given epoch i, the goal is to select a negative sample S_n^i, such that:

$$\cos(g(f(S_a^i)), g(f(S_p^i))) > \cos(g(f(S_a^i)), g(f(S_n^i))) + \alpha_{margin} \tag{4.2.3}$$

$$\tau_{S_n^i} > \tau_{S_n^{i-1}} \tag{4.2.4}$$

Where, S_a^i is the anchor speech sample, S_p^i is the positive speech sample , and α_{margin} is the margin, as also illustrated in Figure 4.3. Here, $\tau_{S_n^i}$ is a parameter that denotes the average difficulty of S_n^i (a negative sample), chosen at epoch i. The difficulty of a negative sample is computed using its cosine similarity to the corresponding anchor speech sample in the triplet. Harder negative samples typically have higher cosine similarity to the corresponding anchor samples, making them harder to separate from the anchor samples. A value of $\tau = 0$ yields the easiest negative sample and $\tau = 1$ yields the hardest negative sample, as shown in Figure 4.3. In our experiments, the value of τ is determined by the current stage (or epoch) of the training process. We initialize the training with the value of τ at 0.4 (empirically chosen) and increase it gradually to 1.0 through the course of the

training. This is done to ensure a minimum difficulty of the training triplets at the beginning of the training which is gradually increased as the training proceeds. This helps in avoiding the problem of bad local minima caused by introducing harder negative triplets directly at the beginning of the training [147]. It is also observed that learning only on easy and semi-hard triplets lead to poor generalization capability of the model on harder evaluation pairs. Additionally, the model is pre-trained in the identification mode to ensure easier initialization of the training process.

4.2.2 Analysis of the Proposed DeepVOX Architecture

In Section 4.2.1.3, we introduced our proposed DeepVOX architecture for extracting short-term speech features. In this section, we mathematically analyze the proposed architecture and compare the feature learning process of our proposed algorithm with some popular short-term spectral feature extraction algorithms such as MFCC, PNCC, PLP and MHEC.

However, before proceeding with the mathematical analysis of the proposed DeepVOX network architecture, we first draw a visual comparison between some of the most popular short-term spectral feature extraction algorithms in Figure 4.4. The main purpose of this comparison is to identify the building blocks of different short-term spectral features and develop an understanding of their individual roles in the feature extraction process. Different short-term spectral feature extraction algorithms process speech data differently but they still share some common design elements indicated by same-colored outlines in Figure 4.4. We further use this comparative study to explain the similarities and dissimilarities between our proposed algorithm and some of the existing short-term spectral feature extraction algorithms.

Furthermore, please note that the DeepVOX method is proposed as an alternative for short-term spectral features such as MFCC and LPC and is intended to be used alongside feature embedding methods such as xVector, iVector, or 1D-Triplet-CNN for performing speaker recognition. Therefore, DeepVOX is strictly a short-term time-domain feature extraction method, whereas xVector, iVector, and 1D-Triplet-CNN are speech feature embedding methods. The DeepVOX method, similar to MFCC and LPC, extracts variable-length short-term time-domain features for an input raw

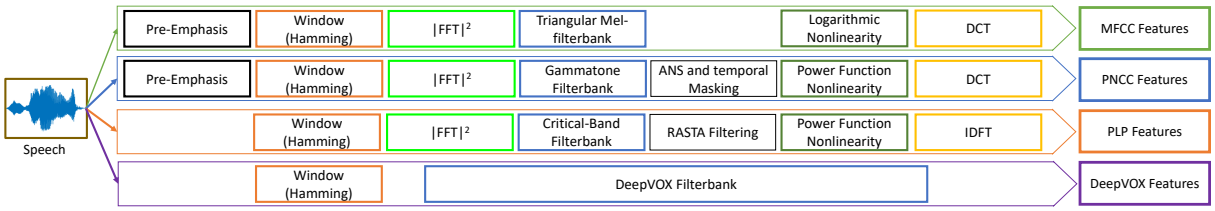


Figure 4.4: A visual comparison of different Short-term spectral feature extraction algorithms with our proposed DeepVOX algorithm. Boxes outlined in same colors perform similar types of operations in the corresponding feature extraction processes.

speech audio, i.e., a 160XN dimensional input speech frame yields a 40XN dimensional DeepVOX feature. Here, N depends on the length of the audio. In contrast, speech embedding techniques such as xVector, iVector, and 1D-Triplet-CNN extract the speaker-dependent features from a variable-length MFCC feature input into a single fixed-dimensional embedding. Additionally, DeepVOX features, unlike the xVector embeddings, are not a mid-level representation drawn from an end-to-end speaker recognition neural network. Instead, DeepVOX is an independent neural network model carefully designed to learn a time-domain speech filterbank directly from raw audio data. Such an approach makes the DeepVOX features, unlike existing deep learning-based speech embedding networks [83, 84, 154], a direct alternative for short-term spectral features such as MFCC and LPC in a wide variety of speaker recognition models. We specifically trained xVector and iVector models using DeepVOX features, described in Section 7.4.2 to demonstrate its compatibility with existing deep learning-based and classical speaker recognition methods. The experimental results given Section 4.4 show its performance benefits over MFCC, LPC, and MFCC-LPC features.

4.2.2.1 Building Blocks of Short-term Spectral Feature Extraction Algorithms

The comparison in Figure 4.4 highlights some key components, given below, important for designing a short-term spectral feature extraction algorithm.

- Pre-emphasis: In the pre-emphasis phase, the speech signal is passed through a high-pass filter to compensate for the natural suppression of high frequency components in the sound production apparatus of humans. This step amplifies the higher-frequency formants and makes the speech

sound sharper. Since, this step can have a negative effect on the quality of speech if the input audio has high-frequency noise artifacts, we decided to skip this phase in our proposed algorithm.

• Framing and Windowing: In the framing phase, the speech signal is split into smaller short-term audio frames, typically 20-30ms long. This is done to reliably extract speaker-dependent vocal characteristics, which are stable only within such short-term frames. We use a frame-length of 20ms and a stride of 10ms for slicing the speech signal into frames. In the windowing phase, the short-term frames are usually multiplied by a window function, such as hamming window in our case, for making the start and end of the short-time audio frames continuous.

• Fourier Transform: FFT (Fast Fourier Transform) is performed to decompose a speech signal based on its frequency content. Usually only the magnitude of the frequency response is used in the feature extraction process. However, as previously discussed, phase information of the frequency response can also be used alongside the magnitude to further improve the performance of speaker recognition systems. Alternatives to FFT-based signal decomposition such as non-harmonic bases, aperiodic functions and data-driven bases derived from independent component analysis (ICA) have been studied in literature [179]. Instead of separating the different sounds in our speech frames into frequency components using FFT, the proposed DeepVOX network learns speech features in the time domain itself.

• Filterbank Integration: The FFT magnitude response is then processed through filterbanks of different shapes such as triangular, rectangular, etc. and placed on different scales such as Mel-scale and Bark-scale. Mostly the choice of filterbanks is driven by psychoacoustic studies involving human hearing and perception [153, 176]. Mel frequency-bank and Gammatone frequency-bank are two such examples of handcrafted filterbanks used in MFCC and PNCC features respectively. For DeepVOX the goal is to learn data-driven filterbanks which are non-linear combination of multiple convolutional filters and are specifically suited for performing speaker recognition.

• Nonlinear Rectification: This step is done to compress the dynamic range of filterbank energies. The importance of this step is demonstrated in [176] where replacing the logarithmic nonlinearity

with cubic root, due to its robustness to audio degradations, lead to improved speaker recognition performance. However, for the DeepVOX there is no need for an explicit non-linear rectification step due to the inherent non-linearity in the network architecture.

4.2.2.2 Mathematical Analysis of the DeepVOX Architecture

Majority of the popular short-term spectral feature extraction algorithms such as MFCC, PNCC, etc. extract the speaker dependent features from a speech signal using pre-defined filterbanks in spectral domain. To this effect, the Fourier Transform is used to decompose a speech signal into its constituent frequencies, thereby, making filtering operation semantically easier. Additionally, from the implementation perspective, the filtering operation in the Fourier domain is computationally cheaper than in time domain. This is because, as per the convolution theorem, the computationally-expensive convolution operation, between the signal and the filter, in time domain is replaced by pointwise multiplication in the fourier domain. The Fourier Transform is usually implemented using the Fast Fourier Transform (FFT) algorithm which makes the filtering of 1D audio signals even more computationally efficient, $\mathscr{O}(n \log n)$, as compared to performing general convolution operation, $\mathscr{O}(n^2)$. However, FFT only provides a close approximation of time domain filtering and is often inconsistent across different implementations of the FFT algorithm [145], thereby enforcing a trade-off between computational complexity and accuracy. The computational complexity of convolution operations in time domain filtering initially made it inefficient for practical implementation. However, the recent development of extremely efficient implementations and dedicated hardware for the convolution operation makes Convolutional Neural Networks (CNN) extremely well-suited for performing time domain filtering. Therefore, we use CNN in our algorithm to learn time-domain filters efficiently from raw speech audio.

As discussed earlier and illustrated in Figures 4.1 and 7.1, our proposed DeepVOX architecture takes a 2D *speech frame* S derived from raw speech waveform, as input to the network. A speech frame S can be represented as:

$$S = [u_1, u_2, \cdots, u_i, \cdots, u_n] \tag{4.2.5}$$

Where u_i is the i^{th} speech unit in the speech frame S and n is the total number of speech units in a speech frame. As per the design of the DeepVOX architecture, the network outputs a 40 channel filter response f_i corresponding to speech unit u_i in a speech frame S. Therefore, the output $\mathbf{O}$ of the DeepVOX can be given by:

$$\mathbf{O} = [f_1, f_2, \cdots, f_i, \cdots, f_n] \tag{4.2.6}$$

Where, f_i is given by:

$$f_i = \Big[x_{i,1}, x_{i,2}, \ldots, x_{i,j}, \ldots, x_{i,40}\Big]^\top \tag{4.2.7}$$

Here, $x_{i,j}$ is the j^{th} channel filter output for i^{th} speech unit u_i. In the DeepVOX model, channel outputs at the final layer are results of multiple convolutions of the input data with different convolution filters across the depth of the network. Therefore, the network output f_i corresponding to speech unit u_i can be written as:

$$f_i = (l_m(l_{m-1}(\cdots l_k(\cdots l_1(u_i)) \tag{4.2.8}$$

Here, $l_k()$ is the k^{th} layer output of the DeepVOX model and m is the total number of layers. Each layer of DeepVOX learns a multi-channel convolutional filter C_k. We can represent $l_k()$ as:

$$l_k(u_i) = C_k \circledast u_i, \tag{4.2.9}$$

where C_k is the convolutional filter for the k^{th} layer. The operation in the eq. 4.2.9 is equivalent to time-domain filtering of input signal u_i with filter C_k. Hence, we can rewrite the eq. 4.2.8 as:

$$f_i = (C_m \circledast (C_{m-1} \circledast (\cdots C_k \circledast (\cdots C_1 \circledast (u_i)), \tag{4.2.10}$$

Since, the convolution operation is associative, we can rewrite eq. 4.2.10 as:

$$f_i = \underbrace{(C_m \circledast C_{m-1} \circledast \cdots \circledast C_k \circledast \cdots C_1)}_{\text{learned DeepVOX filterbank}} \circledast u_i \tag{4.2.11}$$

$$DeepVOX_{filterbank} = C_m \circledast C_{m-1} \circledast \cdots \circledast C_k \circledast \cdots C_1 \tag{4.2.12}$$

The $DeepVOX_{filterbank}$, therefore, is designed to learn a 40 channel convolution filter through a combination of multi-channel time-domain filters learned in different layers of the DeepVOX model. Here, each of the 40 channels represents an individual time-domain speech filter in the $DeepVOX_{filterbank}$.

Table 4.1: Verification Results on the VOXCeleb2 speech dataset. The proposed DeepVOX features outperform the baseline features for majority of the speaker recognition algorithms, across all the metrics.

#	Method	TMR@FMR={1%, 10%}				minDCF (ptar={0.001, 0.01})				EER(in %)			
		MFCC	LPC	MFCC-LPC	Deep VOX	MFCC	LPC	MFCC-LPC	Deep VOX	MFCC	LPC	MFCC-LPC	Deep VOX
1	1D-Triplet-CNN-online	70.72, 93.13	78.05, 94.93	82.09, 97.55	**91.98, 98.45**	0.080, 0.67	0.067, 0.58	0.062, 0.43	**0.030, 0.28**	8.42	6.84	5.42	**2.92**
	1D-Triplet-CNN	69.30, 93.5	74.33, 94.57	84.70, 95.77	**90.49, 98.09**	0.078, 0.63	0.077, 0.54	0.075, 0.45	**0.045, 0.37**	8.62	7.06	6.05	**3.46**
	xVector-PLDA	55.75, 85.96	73.61, 95.07	76.76, 94.75	**90.76, 97.69**	0.080, 0.78	0.074, 0.54	0.072, 0.52	**0.048, 0.37**	11.25	7.35	7.35	**3.95**
	iVector-PLDA	86.16, 96.02	81.57, 97.1	92.54, **98.29**	**93.72**, 98.14	**0.050**, 0.34	0.078, 0.53	0.056, **0.32**	0.063, 0.39	5.39	6.32	**3.37**	3.63
	RawNet2	91.75, 97.48				0.056, 0.30				3.91			

Table 4.2: Verification Results on the degraded Fisher speech dataset. The proposed DeepVOX features outperform the baseline features for a majority of methods and data partitions, across all the metrics.

#	Training set / Testing set	Method	TMR@FMR={1%, 10%}				minDCF (ptar={0.001, 0.01})				EER(in %)			
			MFCC	LPC	MFCC-LPC	Deep VOX	MFCC	LPC	MFCC-LPC	Deep VOX	MFCC	LPC	MFCC-LPC	Deep VOX
2	F1/F1	M1	49.13, 82.06	46.60, 81.87	59.93, 87.46	**79.14**, **93.05**	0.089, 0.89	0.094, 0.87	0.081, 0.81	**0.075, 0.52**	13.86	14.05	11.82	**7.99**
		M2	27.98, 74.62	31.64, 84.81	51.81, 84.81	**77.27**, **92.53**	0.095, 0.95	0.094, 0.93	0.087, 0.83	**0.051, 0.51**	16.50	17.06	12.65	**8.30**
		M3	20.77, 57.93	20.58, 63.22	29.10, 72.61	**53.31, 88.63**	0.097, 0.97	0.097, 0.97	0.096, 0.96	**0.089, 0.87**	22.86	20.43	17.46	**10.92**
		M4	25.42, 68.32	03.40, 18.01	29.04, 70.66	**71.12, 90.23**	0.098, 0.97	0.099, 0.99	0.096, 0.96	**0.074, 0.63**	18.47	43.58	18.13	**9.77**
		M5	62.53, 84.50				0.084, 0.65				13.61			
3	F1 / F2	M1	28.36, 71.49	27.15, 63.86	39.73, 77.98	**78.51, 93.13**	0.094, 0.94	0.095, 0.95	0.091, 0.91	**0.091, 0.53**	17.75	20.77	15.72	**7.99**
		M2	14.35, 55.44	9.18, 46.56	34.74, 74.09	**75.73, 92.33**	0.098, 0.98	0.099, 0.99	0.094, 0.94	**0.056, 0.49**	23.30	25.98	17.37	**8.42**
		M3	12.65, 46.68	2.98, 18.84	**12.27, 53.02**	7.90, 36.98	0.099, 0.99	**0.098, 0.98**	0.099, 0.99	0.099, 0.99	26.59	44.3	24.02	**31.3**
		M4	5.41, 25.10	11.58, 42.21	14.78, 54.10	**18.63, 55.50**	0.097, 0.97	0.100, 0.99	0.099, 0.99	**0.096, 0.96**	37.87	30.93	23.54	**26.10**
		M5	27.93, 59.75				0.094, 0.93				27.53			
4	F2 / F2	M1	47.62, 83.12	46.22, 82.21	55.78, 86.97	**80.25, 94.08**	0.081, 0.81	0.087, 0.84	0.085, 0.83	**0.062, 0.57**	13.37	14.24	11.56	**7.25**
		M2	36.40, 77.49	33.42, 76.02	50.57, 84.67	**75.13, 92.65**	0.099, 0.97	0.092, 0.92	0.088, 0.88	**0.081, 0.74**	16.16	16.43	13.03	**8.54**
		M3	20.77, 57.93	20.58, 63.22	29.10, 72.61	**47.91, 82.00**	0.098, 0.98	0.094, 0.94	0.097, 0.96	**0.096, 0.86**	22.86	20.43	17.46	**13.9**
		M4	16.19, 56.57	19.31, 56.84	29.37, 73.79	**79.22, 92.8**	0.097, 0.96	0.099, 0.99	0.095, 0.95	**0.084, 0.61**	24.08	23.62	16.65	**7.9**
		M5	69.92, 85.85				**0.066, 0.54**				12.52			
5	F2 / F1	M1	20.35, 63.18	19.79, 53.10	34.71, 71.75	**47.56, 86.53**	0.095, 0.95	0.097, 0.97	0.098, 0.96	**0.098, 0.94**	21.26	25.57	19.95	**11.91**
		M2	10.57, 39.80	6.80, 36.18	18.16, 62.31	**45.93, 86.17**	0.100, 0.99	0.099, 0.99	0.099, 0.99	**0.099, 0.90**	30.97	31.76	22.85	**12.18**
		M3	7.61, 29.29	7.04, 28.83	**9.51, 44.39**	6.98, 31.19	0.099, 0.99	0.099, 0.99	0.099, 0.99	**0.097, 0.97**	37.39	31.57	**27.23**	36.59
		M4	11.03, 36.78	3.25, 22.58	**11.71, 41.62**	3.89, 37.74	**0.098, 0.98**	0.099, 0.99	0.099, 0.99	0.100, 0.99	31.46	41.35	29.00	**25.6**
		M5	23.75, 66.18				0.0100, 1.00,				22.32			

Method	M1	M2	M3	M4	M5
Algorithm	1D-Triplet-CNN-online	1D-Triplet-CNN	xVector-PLDA	iVector-PLDA	RawNet2

Data Subset	F1	F2
Noise Characteristics	Babble, R1,V1	F16, R1, V1

4.3 Datasets and Experiments

In this work, we perform multiple speaker verification experiments on a variety of datasets and protocols. Primarily, we use the following datasets for training and evaluating the proposed and baseline speaker verification algorithms.

1. VOXCeleb2 dataset [43]

2. Fisher English Training Speech Part 1 dataset [44]

3. NIST SRE 2008 [1], 2010 [2], and 2018 [3] datasets

We also create degraded versions of the Fisher and NIST SRE 2008 speech datasets by adding different types of noise data from the NOISEX-92 [166] dataset under varying levels of (signal-to-noise ratio) SNR (0 to 20 dB) and reverberations. This is done to evaluate the robustness of our

Table 4.3: Verification Results on the original and degraded, NIST SRE 2008, 2010, and 2018 datasets. The proposed DeepVOX features outperform the baseline features for a majority of methods and data partitions, across all the metrics.

#	Train set / Test set	Method	TMR@FMR={1%, 10%}				minDCF (ptar={0.001, 0.01})				Equal Error Rate (EER, in %)			
			MFCC	LPC	MFCC-LPC	Deep VOX	MFCC	LPC	MFCC-LPC	Deep VOX	MFCC	LPC	MFCC-LPC	Deep VOX
6	P1 / P1	M1	55.21, 93.06	41.49, 87.25	52.50, 93.22	**81.05, 97.63**	0.097, 0.76	0.084, 0.84	0.095, 0.89	**0.081, 0.60**	8.74	11.18	8.18	**4.45**
		M2	53.17, 89.12	49.17, 86.65	60.21, 93.36	**81.37, 97.30**	0.082, 0.82	0.085, 0.83	0.079, 0.76	**0.066, 0.59**	10.55	11.62	8.34	**4.77**
		M3	25.20, 78.60	22.96, 76.47	**24.00, 85.21**	23.97, 78.72	0.099, 0.99	0.098, 0.98	**0.098, 0.98**	0.099, 0.99	14.15	15.15	**11.95**	14.68
		M4	48.70, 85.13	30.64, 78.20	42.16, 88.35	**37.63, 96.12**	**0.087, 0.87**	0.097, 0.97	0.093, 0.93	0.094, 0.93	12.37	15.85	10.81	**6.85**
		M5	81.62, 93.57				0.047, 0.47				7.53			
7	P1 / P2	M1	8.40, 24.93	7.58, 23.56	**8.40, 24.47**	4.84, 21.00	0.096, 0.96	0.098, 0.98	**0.096, 0.96**	0.098, 0.98	**43.29**	43.65	43.74	47.31
		M2	2.28, 21.64	2.65, 18.54	4.13, **25.20**	**6.57**, 23.19	0.099, 0.99	0.099, 0.99	0.099, 0.99	**0.098, 0.98**	45.02	44.11	**39.40**	46.57
		M3	3.01, **19.27**	1.74, 15.62	2.10, 17.17	**4.01,** 19.17	0.099, 0.99	0.099, 0.99	0.099, 0.99	**0.097, 0.97**	**43.84**	46.39	45.57	46.66
		M4	3.29, 16.35	3.74, 17.26	1.19, 10.14	**3.37, 19.54**	**0.098, 0.98**	0.099, 0.99	0.099, 0.99	0.099, 0.99	44.75	**44.29**	47.40	46.30
		M5	0, 15.35				0.100, 1.00				44.46			
8	P1 / P3	M1	9.92, 32.07	6.73, 24.73	**10.46, 32.09**	8.06, 29.53	0.099, 0.99	0.099, 0.99	**0.098, 0.98**	0.099, 0.99	38.95	42.39	**38.43**	39.04
		M2	8.45, 29.69	5.74, 22.99	**9.75, 30.17**	6.73, 26.27	0.099, 0.99	0.099, 0.99	0.099, 0.99	**0.099, 0.99**	**38.98**	42.67	39.78	40.30
		M3	1.89, 15.44	1.47, 12.02	1.34, 13.95	**4.41, 19.14**	0.099, 0.99	0.099, 0.99	0.100, 1.00	**0.099, 0.99**	45.32	48.30	46.63	**45.24**
		M4	5.35, 24.57	1.02, 12.04	4.18, 20.64	**5.72, 24.57**	0.099, 0.99	0.099, 0.99	0.100, 1.00	**0.099, 0.99**	**40.16**	47.98	42.32	41.20
		M5	2.50, 21.54				0.100, 1.00				41.36			
9	P4 / P4	M1	35.28, 83.49	38.01, 81.19	35.25, 86.86	70.16, 94.46	0.088, 0.88	0.090, 0.90	0.096, 0.96	**0.058, 0.58**	12.47	13.44	11.40	7.44
		M2	39.28, 84.26	35.48, 80.49	53.92, 90.00	**69.22, 95.36**	0.090, 0.90	0.097, 0.94	0.075, 0.75	**0.073, 0.68**	12.94	14.24	10.00	**7.10**
		M3	22.44, 75.09	20.81, 65.42	23.64, 72.66	**24.17, 63.72**	0.099, 0.99	**0.095, 0.95**	0.099, 0.99	0.099, 0.99	**15.24**	19.24	16.17	21.19
		M4	**39.57**, 82.87	31.58, 72.46	11.70, 41.25	31.30, **83.67**	0.099, 0.99	**0.093, 0.93**	0.099, 0.99	0.099, 0.99	13.53	17.34	28.34	**12.31**
		M5	67.85, 89.68				0.091, 0.66				10.24			
10	P5 / P5	M1	26.70, 68.28	22.21, 61.86	20.01, 59.52	**62.40, 95.19**	0.097, 0.97	0.098, 0.98	0.093, 0.93	**0.080, 0.80**	19.63	21.24	22.64	**7.25**
		M2	35.34, 75.31	29.39, 73.41	43.02, 84.97	**71.36, 94.68**	0.097, 0.97	0.095, 0.95	0.092, 0.89	**0.067, 0.64**	16.29	17.19	12.67	**6.99**
		M3	17.15, 58.77	17.58, 54.97	22.03, 66.63	**36.20, 77.43**	0.096, 0.96	0.097, 0.97	0.098, 0.98	**0.084, 0.84**	20.88	22.28	19.27	**15.57**
		M4	22.73, 60.57	6.10, 28.74	4.45, 23.00	**27.30, 86.43**	**0.095, 0.95**	0.098, 0.98	0.099, 0.99	0.099, 0.99	21.13	36.96	37.89	**11.15**
		M5	63.15, 90.81				0.071, 0.71				9.50			
11	P4 / P5	M1	8.00, 34.59	9.65, 36.92	8.83, 38.86	**15.46, 58.06**	0.099, 0.99	**0.098, 0.98**	0.099, 0.99	0.099, 0.99	31.97	33.55	29.49	**22.46**
		M2	14.42, 49.12	14.78, 47.04	**18.41, 55.36**	11.37, 47.75	0.099, 0.99	0.099, 0.99	**0.097, 0.97**	0.099, 0.99	26.01	28.13	**23.29**	26.08
		M3	7.71, 31.97	8.22, 35.06	14.53, **53.00**	**15.97**, 40.98	0.097, 0.97	0.099, 0.99	**0.096, 0.96**	0.099, 0.99	34.95	31.43	**22.46**	31.83
		M4	6.03, 27.92	3.70, 20.85	2.22, 15.97	**6.09**, **28.34**	0.099, 0.99	0.099, 0.99	0.099, 0.99	**0.099, 0.99**	35.24	41.51	43.24	**34.76**
		M5	13.85, 47.32				0.099, 0.99				25.97			
12	P5 / P4	M1	19.14, 58.55	7.10, 40.01	19.14, 58.55	**35.05, 78.74**	0.0947, 0.94	0.0995, 0.99	0.0986, 0.98	**0.0945, 0.94**	22.67	28.74	22.67	**15.22**
		M2	11.34, 37.08	4.57 , 27.84	19.34, 56.59	**21.09, 68.32**	0.0972, 0.97	0.0998, 0.99	**0.0972, 0.97**	0.0976, 0.97	32.28	37.55	23.61	**18.29**
		M3	12.17, 45.38	12.77, 52.82	**14.54, 47.35**	12.98, 40.42	0.0999, 0.99	0.0986, 0.98	0.0988, 0.98	**0.0981, 0.98**	**27.54**	22.87	27.64	31.01
		M4	**9.50**, 36.15	3.60, 21.51	3.33, 20.21	7.54, **37.95**	**0.0990, 0.99**	0.0995, 0.99	0.0999, 0.99	0.0997, 0.99	34.11	40.88	41.71	**32.0**
		M5	9.04, 41.75				0.100, 0.99				27.16			

Method	M1	M2	M3	M4	M5
Algorithm	1D-Triplet-CNN-online	1D-Triplet-CNN	xVector-PLDA	iVector-PLDA	RawNet2

Data Subset	P1	P2	P3	P4	P5
Noise Type	NIST SRE 08	NIST SRE 10	NIST SRE 18	P1 + Babble	P1 + F16

Table 4.4: Verification Results on multi-lingual speakers from the NIST SRE 2008 dataset. The proposed DeepVOX features outperform the baseline features for a majority of methods and data partitions, across all the metrics.

#	Train set / Test set	Method	TMR@FMR={1%, 10%}				minDCF (ptar={0.001, 0.01})				Equal Error Rate (EER, in %)			
			MFCC	LPC	MFCC-LPC	Deep VOX	MFCC	LPC	MFCC-LPC	Deep VOX	MFCC	LPC	MFCC-LPC	Deep VOX
13	L1 / L1	M1	47.88, 85.30	45.26, 85.26	55.94, 90.34	**80.30, 99.16**	0.095, 0.89	0.088, 0.85	0.092, 0.85	**0.062, 0.56**	11.90	12.58	9.80	**3.98**
		M2	33.44, 79.70	36.34, 77.88	47.54, 86.70	**77.60, 99.30**	0.094, 0.91	0.089, 0.89	0.093, 0.90	**0.075, 0.63**	13.92	14.78	11.30	**4.32**
		M3	47.88, 85.30	45.26, 85.26	55.94, 90.34	**72.84, 97.94**	0.090, 0.90	0.091, 0.87	0.090, 0.81	**0.089, 0.66**	11.90	12.58	9.80	**5.64**
		M4	46.86, 83.58	41.46, 83.24	60.06, 93.76	**76.54, 98.42**	0.094, 0.88	0.098, 0.87	0.078, 0.75	**0.089, 0.65**	12.74	12.96	8.14	**5.00**
		M5	71.54, 95.64				0.084, 0.75				6.86			
14	L1 / L2	M1	39.52, 82.03	43.40, 79.60	47.95, 86.53	**77.26, 97.87**	0.096, 0.88	0.089, 0.86	0.083, 0.78	**0.063, 0.60**	13.56	14.7	11.61	**5.04**
		M2	32.39, 74.86	35.80, 75.04	41.67, 83.09	**66.91, 97.70**	0.097, 0.97	0.095, 0.91	0.089, 0.84	**0.075, 0.64**	16.21	16.77	13.1	**5.17**
		M3	39.52, 82.03	43.40, 79.60	47.90, 86.50	**72.49, 97.57**	0.095, 0.92	0.094, 0.83	0.090, 0.90	**0.079, 0.66**	13.56	14.7	11.61	**5.96**
		M4	40.48, 80.17	39.58, 78.17	56.23, 88.30	**77.64, 98.39**	0.098, 0.96	0.085, 0.85	0.090, 0.78	**0.061, 0.55**	14.1	15.02	10.74	**4.78**
		M5	67.30, 93.18				0.091, 0.69				8.03			
15	L1 / L3	M1	29.06, 70.46	28.10, 64.68	33.14, 74.82	**62.24, 88.82**	0.095, 0.94	0.098, 0.97	0.092, 0.90	**0.081, 0.74**	17.64	21.26	16.52	**10.72**
		M2	25.78, 64.28	18.38, 57.04	30.82, 67.60	**55.96, 89.02**	0.097, 0.97	0.098, 0.98	0.094, 0.92	**0.098, 0.88**	20.30	23.04	18.80	**10.60**
		M3	29.06, 70.46	28.10, 64.68	47.95, 86.53	**54.42, 87.88**	0.093, 0.93	0.097, 0.97	0.094, 0.94	**0.091, 0.84**	17.64	21.26	11.61	**11.20**
		M4	26.30, 66.30	20.72, 61.40	38.70, 74.80	**56.90, 88.06**	0.094, 0.94	0.096, 0.96	0.092, 0.89	**0.098, 0.86**	19.52	22.00	16.86	**11.16**
		M5	50.40, 81.44				0.090, 0.85				14.58			

Method	M1	M2	M3	M4	M5
Algorithm	1D-Triplet-CNN-online	1D-Triplet-CNN	xVector-PLDA	iVector-PLDA	RawNet2

Data Subset	L1	L2	L3
Language Characteristics	English Only	Multi-Lingual	Cross-Lingual

proposed method to a wide variety of audio degradations. Additionally, all the speech datasets were sampled at a rate of $8,000$Hz to match the NIST SRE dataset specifications [1]. We also perform speaker verification experiment on speech samples of varying audio lengths, as also done in [42]. This experiment is important for evaluating the dependence of a speaker recognition algorithm on

the duration of speech audio available for evaluation. As in practice, the duration of usable speech audio available for evaluation is often limited and is further reduced by degradations.

Table 4.5: Verification Results under varying audio length on the NIST SRE 2008 dataset. The proposed DeepVOX features outperform the baseline features for a majority of methods and data partitions, across all the metrics.

Length (secs)	Method	TMR@FMR={1%, 10%}				minDCF (ptar={0.001,0.01})				Equal Error Rate (EER, in %)			
		MFCC	LPC	MFCC-LPC	Deep VOX	MFCC	LPC	MFCC-LPC	Deep VOX	MFCC	LPC	MFCC-LPC	Deep VOX
3.5	M1	55.20, 93.05	42.28, 86.84	49.43, 92.32	**80.59, 97.63**	0.094, 0.78	0.087, 0.85	0.090, 0.83	**0.079, 0.62**	8.74	11.61	8.57	**4.52**
	M2	59.61, 90.72	52.67, 88.58	65.99, 94.53	**79.87, 97.74**	0.088, 0.72	0.083, 0.79	0.080, **0.69**	**0.076**, 0.71	9.65	10.71	7.64	**4.59**
	M3	27.10, 78.81	19.26, 74.70	24.57, **81.21**	**29.81**, 77.39	0.099, 0.99	0.099, 0.99	**0.097, 0.97**	0.099, 0.99	14.39	15.45	**12.92**	15.24
	M4	44.89, 78.60	25.50, 75.70	37.48, 86.28	**51.34, 95.87**	0.092, 0.92	0.098, 0.98	0.096, 0.96	**0.078, 0.78**	14.82	16.49	11.92	**6.9**
	M5	82.23, 93.86				0.056, 0.47				7.39			
3.0	M1	55.90, 91.02	41.48, 85.14	52.80, 92.15	**80.05, 97.48**	0.093, 0.80	0.089, 0.88	0.094, 0.83	**0.077, 0.62**	9.47	12.04	8.87	**4.73**
	M2	57.58, 90.22	50.63, 88.58	65.49, 94.13	**76.89, 97.74**	**0.075**, 0.74	0.085, 0.77	0.078, 0.70	0.083, **0.64**	9.85	10.75	7.71	**4.63**
	M3	24.63, 76.50	18.46, 71.16	23.66, **79.11**	**28.99**, 75.60	**0.098, 0.97**	0.099, 0.99	0.098, 0.98	0.099, 0.99	15.15	17.12	**14.12**	15.89
	M4	41.62, 77.27	25.03, 71.50	35.11, 84.71	**51.66, 95.19**	0.093, 0.92	0.098, 0.98	0.096, 0.96	**0.080, 0.80**	16.19	17.86	12.65	**7.03**
	M5	81.16, 94.15				0.046, 0.46				7.28			
2.5	M1	54.17, 89.19	41.98, 85.41	54.33, 91.78	**77.11, 97.31**	0.090, 0.82	0.087, 0.87	0.091, 0.78	**0.059, 0.59**	10.04	12.24	9.17	**5.10**
	M2	54.44, 89.95	47.50, 88.15	66.86, 94.23	**74.56, 97.34**	0.080, 0.80	0.081, 0.81	0.086, 0.73	**0.071, 0.61**	10.01	11.11	7.74	**5.10**
	M3	**39.92,** 70.83	20.23, 67.49	31.98, **82.04**	28.88, 72.37	**0.097, 0.97**	0.099, 0.99	0.099, 0.99	0.099, 0.99	17.76	19.93	**13.79**	17.22
	M4	20.46, 69.96	16.79, 66.59	24.13, 75.33	**49.73, 94.90**	0.094, 0.87	0.098, 0.98	0.095, 0.95	**0.079, 0.78**	17.09	18.79	15.32	**7.60**
	M5	77.03, 93.21				0.063, 0.51				8.14			
2.0	M1	51.73, 86.41	42.05, 83.84	51.26, 89.68	**74.74, 96.91**	0.090, 0.80	0.092, 0.87	0.087, 0.84	**0.075, 0.68**	11.34	13.08	10.14	**5.45**
	M2	55.77, 87.98	48.20, 85.78	60.01, 93.16	**71.91, 97.24**	0.085, 0.77	0.085**, 0.72**	0.075, 0.75	**0.075**, 0.75	10.81	12.18	8.28	**5.53**
	M3	17.82, 61.58	13.68, 57.38	20.69, 66.62	**23.28, 68.17**	**0.098, 0.98**	0.098, 0.98	0.099, 0.99	0.099, 0.99	20.46	21.83	**18.32**	19.66
	M4	30.77, 66.99	17.69, 59.78	24.73, 78.14	**44.31, 93.72**	0.097, 0.95	0.097, 0.97	0.097, 0.97	**0.090, 0.89**	20.43	22.50	15.29	**8.14**
		69.86, 89.84				0.068, 0.66				10.08			
1.5	M1	44.89, 82.17	36.21, 77.77	45.52, 86.21	**71.33, 96.30**	0.095, 0.91	0.088, 0.88	0.086, 0.85	**0.085, 0.63**	13.71	15.08	11.71	**6.03**
	M2	45.56, 86.42	49.70, 84.95	56.11, 91.66	**63.08, 96.27**	0.093, 0.88	0.092, 0.85	0.085, 0.79	**0.082, 0.72**	11.75	12.25	9.01	**6.17**
	M3	14.59, 52.00	11.62, 47.80	15.99, 57.01	**17.68, 57.98**	0.098, 0.98	0.099, 0.99	**0.097, 0.97**	0.099, 0.99	24.73	26.30	**22.56**	23.07
	M4	19.13, 58.41	13.35, 49.00	20.33, 68.89	**33.04, 89.91**	0.097, 0.97	0.098, 0.98	0.098, 0.98	**0.092, 0.92**	24.37	27.24	18.42	10.08
	M5	64.15, 86.65				0.083, 0.62				12.05			
1.0	M1	33.74, 70.42	29.00, 69.85	40.02, 79.93	**62.68, 94.40**	0.086, 0.86	0.089, 0.89	0.087, 0.87	**0.078, 0.78**	18.82	18.72	14.51	**7.43**
	M2	39.32, 80.37	35.65, 79.04	50.93, 87.75	**53.35, 94.26**	0.093, 0.91	0.097, 0.95	0.089, 0.87	**0.099, 0.85**	13.72	14.89	11.05	**7.61**
	M3	8.71, 37.51	7.76, 34.75	9.74, 41.20	**11.87, 47.11**	**0.097, 0.97**	0.099, 0.99	0.099, 0.99	0.099, 0.99	31.91	32.66	29.31	**27.77**
	M4	12.92, 40.82	8.31, 33.51	15.65, 54.41	**28.45, 82.31**	0.096, 0.96	0.099, 0.99	0.097, 0.97	**0.096, 0.96**	30.54	33.71	24.33	**12.98**
	M5	44.27, 73.51				0.093, 0.82				18.47			
0.5	M1	18.42, 47.56	18.49, 52.26	22.73, 59.47	**48.22, 87.01**	0.095, 0.95	0.094, 0.94	**0.091, 0.91**	0.094, 0.93	28.13	26.06	23.29	**11.41**
	M2	21.33, 65.02	23.50, 63.05	34.71, 76.37	**47.36, 85.83**	0.098, 0.98	0.099, 0.99	0.095, 0.95	**0.098, 0.94**	20.56	20.66	15.99	**12.27**
	M3	4.48, 19.38	3.50, 20.04	3.73, 20.04	**6.56, 30.35**	0.099, 0.99	0.099, 0.99	0.099, 0.99	**0.099, 0.99**	43.15	42.62	40.80	**35.48**
	M4	4.14, 22.73	3.70, 19.73	7.04, 31.41	**17.54, 55.47**	0.099, 0.99	0.099, 0.99	0.099, 0.99	**0.097, 0.97**	41.72	44.29	35.88	**22.64**
	M5	23.35, 45.35				0.099, 0.99				31.79			

Method	M1	M2	M3	M4	M5
Algorithm	1D-Triplet-CNN-online	1D-Triplet-CNN	xVector-PLDA	iVector-PLDA	RawNet2

Table 4.6: Language-based speaker verification Results (TMT@FMR=1%) on the NIST SRE 2008 dataset. All models were trained using English-only speech data from the training set of NIST SRE 2008 dataset

Language ID	1D-Triplet-CNN(DeepVOX)	1D-Triplet-CNN(MFCC-LPC)	RawNet2	xVector-PLDA(MFCC)	iVector-PLDA(MFCC)
BEN	87.71	48.60	73.74	44.13	74.30
CFR	78.31	22.89	55.42	10.84	34.94
CHN	72.82	36.93	66.90	36.93	47.04
FAR	81.86	83.82	90.20	51.96	71.57
HIN	67.59	57.46	59.12	36.65	54.14
ITA	88.64	68.18	46.59	31.82	53.41
JPN	78.72	45.39	68.93	41.43	47.65
KHM	89.62	39.62	54.72	40.57	47.17
KOR	64.51	42.49	55.44	33.94	43.52
RUS	78.99	58.23	67.47	38.99	60.76
THA	84.58	56.85	66.72	42.29	57.37
VIE	78.05	61.85	71.89	38.55	61.58
WUU	89.47	94.74	94.74	89.47	89.47
YUH	75.22	60.84	73.67	43.14	58.41

4.3.1 Datasets

4.3.1.1 VOXCeleb2 Dataset

The VoxCeleb2 [43] dataset consists of over 1 million utterances extracted from YouTube videos. The videos contain short clips of interview videos of $6,112$ celebrities recorded on a variety of devices and in diverse ambient conditions. The entire VOXCeleb2 dataset contains $145,569$ video samples from $5,994$ celebrities in the training set and $4,911$ videos from the remaining 118 speakers in the evaluation set. However, for keeping the triplet-based training process computationally tractable, we only use speech data from one randomly selected video for each subject. This leads to $5,994$ videos corresponding to $5,994$ celebrities in the training set and 118 videos from the remaining 118 speakers in the evaluation set. For conducting the experiments given in Section 4.3.2, each video in the dataset is processed to extract the speech audio, sampled at 8000Hz, from its audio track. Any extracted speech audio greater than 5 seconds audio duration is split into multiple 5 second long, non-overlapping audio samples.

4.3.1.2 Fisher English Training Speech Part 1 Dataset

The Fisher dataset is one of the larger speech datasets with respect to the number of speakers, thereby serving a good test-bench for evaluating the modeling capacity of our algorithm in presence of a large number of speakers. This dataset primarily contains pair-wise conversational speech data, collected over telephone channels, from a set of around 12000 speakers. Since the amount of speech data per speaker varies in the dataset, in order to ensure data balance across different speakers, we choose to work with a subset of 6991 speakers, each having at least 250 seconds of speech audio, across 50 samples, after performing voice activity detection. Further, a random subset of 4500 speakers is chosen to train the models and the remaining speakers form the testing set.

As mentioned earlier, we have also added the 'F-16' and 'Babble' noise from the NOISEX-92 [166] noise dataset to the Fisher speech dataset. The resultant 'degraded-Fisher' speech dataset

was maintained at a SNR level of 10dB. Apart from the additive noise from NOISEX-92 [166] noise dataset, we also added convolutive noise in form of reverberations to the speech data generated in a simulated cubical room of side length 4m. The experiments for the Fisher dataset, as given in Table 4.2 and Figure 4.5, are designed to test the robustness of the proposed algorithm to generalize successfully across different types of noise profiles, in both *cross-noise* and *same-noise* scenarios. For example, experiments 1 and 3 in Table 4.2, are termed as same-noise experiments, since the training and testing sets are degraded with same type of noise. Conversely, experiments 2 and 4 in Table 4.2, are termed as cross-noise experiments, since the training and testing sets are degraded with different types of noise.

4.3.1.3 NIST SRE 2008, 2010, and 2018 Datasets

The NIST SRE 2008 [1] dataset is a widely popular dataset in the speaker recognition community, as it encompasses the challenges of performing speaker recognition on multilingual speech data captured under varying ambient conditions. The purpose of using NIST SRE 2008 dataset in our experiments, given in Table 4.3 and Figure 4.5, is to evaluate the performance of our proposed algorithm in the presence of multi-lingual data, as cross-lingual speaker recognition [95] is an open challenge in the speaker recognition community. The diverse noise characteristics of the NIST SRE 2008 dataset together with the our self-added noise, as explained later, makes these experiments emulate real-life speaker recognition challenges. For our experiments, we choose a subset of speech data from the 'phonecall' and 'interview' speech types collected under audio conditions labeled as '10-sec', 'long' and 'short2'. The chosen data subset contains speech from 1336 speakers out of which a randomly chosen subset of 200 speakers is reserved for evaluation purposes, while the rest of the data is used for training our models. The NIST SRE 2008 dataset has channel effects, such as telephone channel, already built into the dataset, making the task of speaker recognition harder. Additionally we also add F-16 and Babble noise, at a resultant SNR of 0dB, to the NIST SRE 2008 dataset to vastly increase the difficulty of the task. We also perform cross-dataset speaker verification performance evaluation using speech data from all the speakers

in the evaluation sets of the NIST SRE 2010 [2] and NIST SRE 2018 [3] datasets.

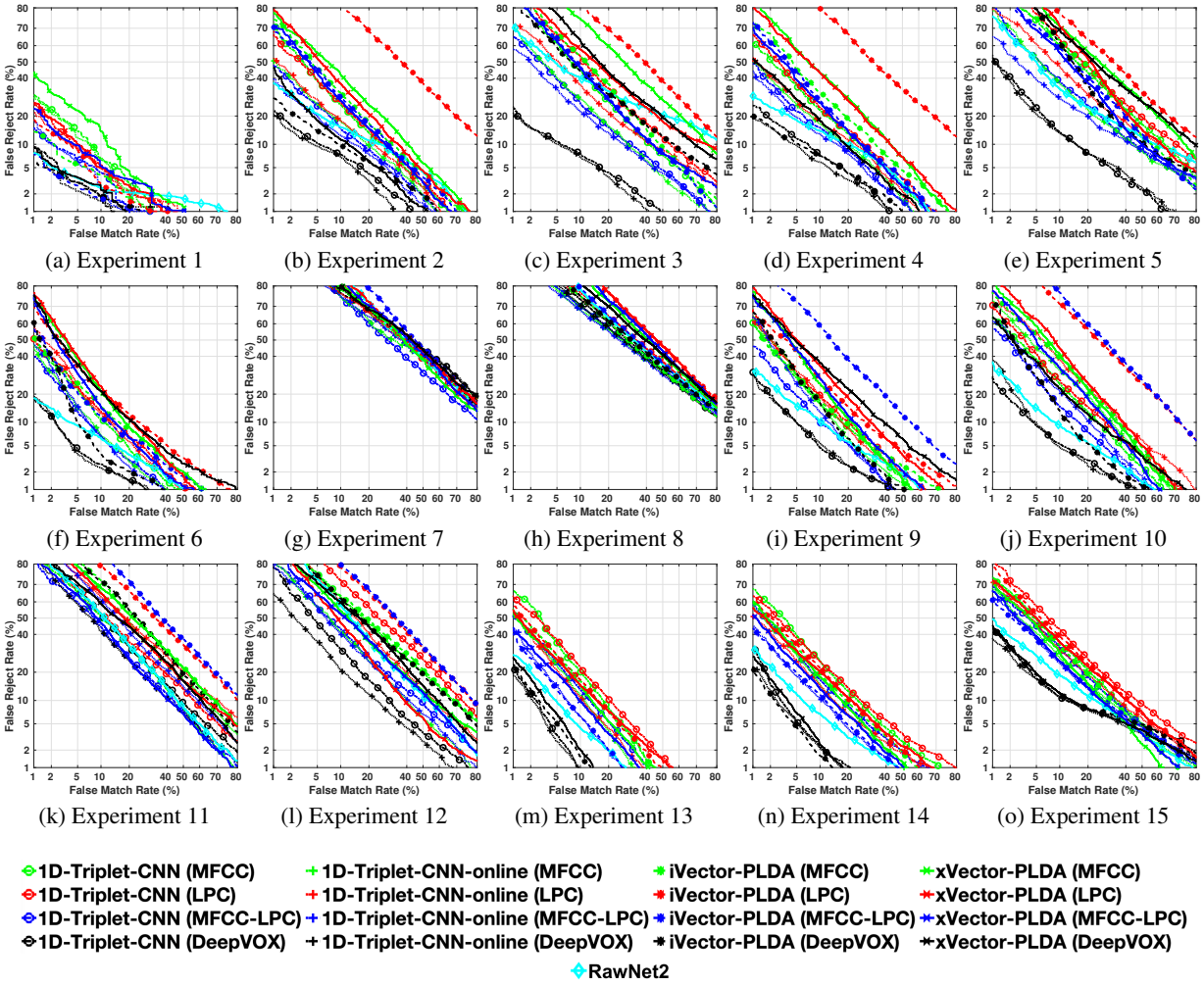
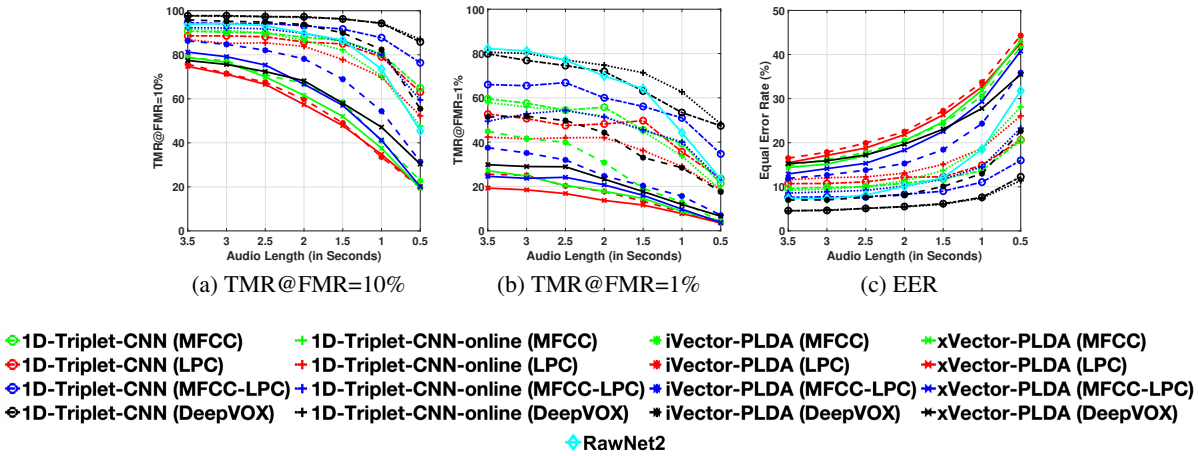


Figure 4.5: DET curves for the speaker verification experiments on the VOXCeleb2 dataset (Exp. 1), degraded Fisher dataset (Exp. 2 to 5, the clean and degraded NIST SRE 2008, 2010, and 2018 datasets (Exp. 6 to 12), and the multilingual subset of NIST SRE 2008 dataset (Exp. 13 to 15) using RawNet2, iVector-PLDA, xVector-PLDA and 1D-Triplet-CNN and 1D-Triplet-CNN-online algorithms on MFCC, LPC, MFCC-LPC, and DeepVOX feature sets.

4.3.2 Experimental Protocols

In all the experiments, we ensure disjoint set of speakers in the training and testing sets. For evaluating robustness of our models we perform same-noise, cross-noise and cross-dataset experiments as shown in Tables 4.1, 4.2, and 4.3. The noise characteristics of the training and testing sets used in the different experiments are given alongside in Tables 4.1, 4.2, and 4.3. For example, in Ex-



(a) TMR@FMR=10% (b) TMR@FMR=1% (c) EER

Figure 4.6: (a) TMR@FMR=10%, (b) TMR@FMR=1%, and (c) EER under varying audio length on the clean NIST SRE 2008 dataset. 1D-Triplet-CNN(DeepVOX) performs the best across varying lengths of test audio.

periment 3 given in Table 4.2, the model was trained on speech data from the training set of Fisher Speech Dataset degraded with Babble noise, and the evaluation was done on speech data from testing set of Fisher Speech Dataset degraded with F16 noise. Note that, no mention of a noise type, such as in Experiment 1 given in Table 4.1, indicates usage of un-altered speech data from the original dataset. Additionally, we have also conducted speaker verification experiments on a subset of multi-lingual speakers from the NIST SRE 2008 dataset, as shown in Table 4.4, for evaluating the effect of speech language on speaker verification performance. Finally, as illustrated in Figure 4.8 and discussed in Section 4.5, we have performed Guided Backpropagation [157] based ablation study of the features extracted by trained DeepVOX models, to understand the type of audio features considered important for performing speaker recognition by the DeepVOX model.

4.3.2.1 Baseline Speaker Verification Experiments

For establishing baseline speaker verification performance on the VOXCeleb2, Fisher, NIST SRE 2008, 2010, and 2018 speech datasets, we choose iVector-PLDA [63] and xVector-PLDA [154] algorithms trained on the baseline features (MFCC, LPC, MFCC-LPC) and DeepVOX features separately. This is done to evaluate and compare the effectiveness of DeepVOX features, with respect to baseline features, in both classical and deep learning-based speaker recognition algo-

rithms. However, unlike the baseline features, DeepVOX feature extraction process requires a DeepVOX model to be trained. For each of the experiments in Tables 4.1, 4.2, and 4.3 we use speech data only from corresponding training set to train the DeepVOX model, ensuring disjoint data and subjects in the training and testing sets for the DeepVOX feature extraction process. We also use the RawNet2 [84] algorithm for establishing baseline raw audio-based speaker recognition performance.

• *iVector-PLDA [63]-based Speaker Verification Experiments*: We conduct experiments using iVector-PLDA as our baseline algorithm. We use speech data from the speakers in training set to train a Universal Background Model (UBM). A total variability (TV) space of 400 dimensions is then learned from the trained UBM. i-vectors are then extracted from the learned total variability (TV) space. A Gaussian-PLDA (gPLDA) model is then trained using the extracted i-vectors. We evaluate the trained model by extracting i-vectors from the speech samples in evaluation pairs. The extracted pairs of i-vectors are then matched using the trained gPLDA model to generate the match scores. We use the MSR Identity Toolkit's [143] implementation of the iVector-PLDA algorithm for conducting our experiments.

• *xVector-PLDA [154]-based Speaker Verification Experiments*: We use the xVector-PLDA algorithm to establish a neural network-based baseline performance for the experiments reported in Tables 4.1, 4.2, 4.3 and 4.5. Since the xVector implementation in the Kaldi [130] toolkit only supports 24-dimensional MFCC features, we use the PyTorch-based implementation of the xVector algorithm [42] due to its compatibility with the 40-dimensional MFCC and LPC features and the 80-dimensional MFCC-LPC features used in our experiments. The PyTorch implementation of the xVector algorithm is paired with a gPLDA based matcher [143] for performing the xVector-PLDA based experiments.

• *RawNet2 [84]-based Speaker Verification Experiments*: We use the RawNet2 algorithm to establish a baseline raw audio-based speaker recognition performance for the experiments reported

in Tables 4.1, 4.2, 4.3 and 4.5. We use the official implementation of the RawNet2 method for performing the RawNet2-based experiments.

4.3.2.2 Speaker Verification Experiments on 1D-Triplet-CNN Algorithm Using MFCC-LPC Feature Fusion

We also perform speaker recognition experiments using the 1D-Triplet-CNN [42] algorithm. These experiments provide benchmark results (given in Tables 4.1,4.2, and 4.3) to directly compare the performance of the DeepVOX feature to MFCC, LPC, and MFCC-LPC features in a deep learning framework. For training the 1D-Triplet-CNN, speech audio triplets are formed using the speakers from the training set. The speech audio triplets are then processed to extract 40 dimensional MFCC and LPC features separately. The extracted MFCC and LPC features are then stacked together to form a two-channel input feature patch for the 1D-Triplet-CNN. For evaluation, speech audio pairs are fed to the trained model to generate pairs of speech embeddings. The speech embeddings are then matched using the cosine similarity metric.

4.3.2.3 Speaker Verification Experiments on 1D-Triplet-CNN Algorithm Using DeepVOX Features (Proposed Algorithm)

In these set of experiments, we evaluate the performance of our proposed approach on multiple training and testing splits (given in the Tables 4.1,4.2, and 4.3) drawn from different datasets and noise types and compare it with the baseline algorithms. Similar to the MFCC-LPC feature-fusion based 1D-Triplet-CNN [42] algorithm, our algorithm also trains on speech audio triplets. However, instead of extracting hand-crafted features like MFCC or LPC, our algorithm trains the DeepVOX and 1D-Triplet-CNN modules together to learn both the DeepVOX-based feature representation and 1D-Triplet-CNN-based speech feature embedding simultaneously. For evaluation, speech audio pairs are fed to the trained DeepVOX model to extract pairs of DeepVOX features which are then fed into the trained 1D-Triplet-CNN model to extract pairs of speech embeddings. The speech embeddings are then matched using the cosine similarity metric.

4.3.2.4 1D-Triplet-CNN-based Speaker Recognition Experiments Using Adaptive Triplet Mining

The proposed adaptive triplet mining technique is evaluated by repeating all the 1D-Triplet-CNN based speaker verification experiments on MFCC, LPC, MFCC-LPC, and DeepVOX features, referred to as *1D-Triplet-CNN-online* in Tables 4.1, 4.2, and 4.3. In our experiments, the 1D-Triplet-CNN models are pretrained in identification mode for 50 epochs followed by 800 epochs of training in verification mode using adaptive triplet mining. As also mentioned in Section 4.2.1.6, the difficulty (τ) of the mined negative samples is gradually increased from 0.4 to 1.0 linearly over 800 epochs. Also, it is important to note that the triplet mining is done in mini-batches of 6 randomly chosen samples drawn from each of the 25 randomly chosen training subjects.

4.3.2.5 Experiments for Studying the Effect of Language on Speaker Verification Performance

The effect of language on speaker recognition performance, also known as the language-familiarity effect (LFE), of both humans and machines, has been studied in the literature [61,99]. According to LFE, human listeners perform speaker recognition better when they understand the language being spoken. Similar trends have been noticed in the performance of automatic speaker recognition systems [99]. In this work, we perform additional speaker recognition experiments (Exp. # 12 to 14 in Table 4.4) on a subset of the NIST SRE 2008 dataset for evaluating the robustness of the DeepVOX features compared to MFCC, LPC, and MFCC-LPC features in the presence of multi-lingual speech data. In all the experiments (Exp. # 12 to 14), the models are trained on English speech data spoken by a subset of 1076 English-speaking subjects in NIST SRE 2008's training set. However, the evaluation sets in experiments 12 to 14 varied as follows:

Same language, english only trials : In Exp. # 12, the trained models are evaluated on same-language (English Only) trials from a subset of 59 multi-lingual subjects in NIST SRE 2008's test set. This experiment establishes the baseline same-language (English to English) speaker verification performance of all the algorithms.

Same language, non-english trials: In Exp. # 13, the trained models are evaluated on same-language (Multi-lingual) trials from a subset of 59 multi-lingual subjects, containing speech data from 15 different languages, in NIST SRE 2008's test set. This experiment aims to investigate the performance of speaker recognition models trained on English-only speech data for matching Non-English same-language (e.g: Chinese to Chinese) speech trials.

Cross-lingual trials: In Exp. # 14, the trained models are evaluated on different-language (Cross-lingual) trials from a subset of 59 multi-lingual subjects, containing speech data from 15 different languages, in NIST SRE 2008's test set. This experiment aims to investigate the performance of speaker recognition models trained on English-only speech data for matching Non-English different-language (e.g., Chinese to Russian) speech trials.

4.3.2.6 Speaker Verification Experiments on Audio Samples of Varying Length

The reliability of the speaker-dependent features extracted from an audio sample depends on the amount of usable speech data present within, which is directly dependent on the length of the audio sample. Therefore, performing speaker recognition in audio samples of a small duration is a challenging task. Since in real-life scenarios, probe audios are of relatively small audio durations (1 sec - 3 secs), the feature extraction algorithm needs to be able to reliably extract speaker-dependent features from speech audio of limited duration. In this experiment (see Table 4.5 and Figure 4.6), we compare the speaker verification performance of our proposed algorithm with the baseline algorithms on speech data of varying duration from the NIST SRE 2008 dataset. The duration of probe audio is varied between 3.5 secs and 0.5 secs in steps of 0.5 secs.

4.4 Results and Analysis

The results for all the experiments described in Section 4.3.2 are given in Tables 4.1, 4.2, 4.3, 4.4, 4.5 and Figures 4.5, 4.6. For all the speaker verification experiments, we report the True Match Rate at False Match Rate of 1% and 10% (TMR@FMR=$\{1\%, 10\%\}$), minimum Detection Cost Function (minDCF) at C_{miss} (cost of a missed detection) value of 1 and Equal Error Rate (EER, in

$\%$) as our performance metrics for comparison of the baseline methods and the proposed method. The minDCF is reported at two different a priori probability of the specified target speaker, P_{tar} of 0.01 and 0.001 (minDCF($P_{tar} = \{0.01, 0.001\}$). The Detection Error Tradeoff (DET) curves are given in Figure 4.5.

• Overall, in all the speaker verification experiments given in Tables 4.1, 4.2, 4.3, 4.4, and 4.5, the 1D-Triplet-CNN algorithm using DeepVOX features trained with adaptive triplet mining, also referred to as 1D-Triplet-CNN-online(DeepVOX), performs the best. The proposed adaptive triplet mining method improves the verification performance (TMR@FMR=1$\%$) of the 1D-Triplet-CNN algorithm using DeepVOX features by 3.01%, and MFCC-LPC features by 8.71%. Similar performance improvements are also noticed for the MFCC and LPC features across all the performance metrics. This establishes the benefits of using the adaptive triplet mining technique over offline-triplet mining for efficiently training the 1D-Triplet-CNN based speaker recognition models.

• Across all the speaker verification experiments given in Tables 4.1, 4.2, 4.3, 4.4, and 4.5, the second-best performance, after DeepVOX features, is obtained by the feature level combination of MFCC and LPC features, referred to as MFCC-LPC features. Therefore, we choose MFCC-LPC features as our strongest baseline feature. In the upcoming discussions, all performance improvements offered by the DeepVOX features, for any particular algorithm, is reported in comparison to the MFCC-LPC features. Furthermore, we will also draw comparison with the RawNet2 model to establish DeepVOX's performance benefits over current state-of-the art raw speech audio-based speaker recognition method.

• In the speaker verification experiment (Exp. #1) on the VOXCeleb2 dataset, given in Table 4.1 and Figure 4.5, the 1D-Triplet-CNN-online(DeepVOX) method performs the best across all the performance metrics. The DeepVOX features improve the speaker verification performance (TMR@FMR=$\{1\%, 10\%\}$), specifically for the 1D-Triplet-CNN-online algorithm, over the best performing baseline feature (MFCC-LPC) by $9.89\%, 0.9\%$. It also reduces the EER by 2.5% and minDCF ($P_{tar} = \{0.001, 0.01\}$) by $\{0.03, 0.15\}$. Similarly, for the 1D-Triplet-CNN algorithm, the

DeepVOX features improve speaker verification performance (TMR@FMR=$\{1\%, 10\%\}$) over the best performing baseline feature (MFCC-LPC) by $5.79\%, 2.39\%$, reduces the EER by 2.58%, and minDCF($P_{tar} = \{0.001, 0.01\}$) by $\{0.03, 0.08\}$. For the xVector-PLDA algorithm, the DeepVOX features improve speaker verification performance (TMR@FMR=$\{1\%, 10\%\}$) over the best performing baseline feature (MFCC-LPC) by $14\%, 2.94\%$, reduces the EER by 3.4% and minDCF($P_{tar} = \{0.001, 0.01\}$) by $\{0.024, 0.15\}$. However, for the iVector-PLDA algorithm, the DeepVOX features exhibit comparable performance to the MFCC-LPC features and vastly outperform the MFCC and LPC features. The 1D-Triplet-CNN(DeepVOX) method also outperforms the RawNet2 across all the performance metrics. The TMR@FMR=$\{1\%, 10\%\}$ is increased by $0.23\%, 0.97\%$, EER is reduced by 0.99%, and minDCF($P_{tar} = \{0.001, 0.01\}$) is reduced by $\{0.026, 0.02\}$.

- In all the four speaker verification experiments (Experiments 2 to 5) on the degraded Fisher dataset given in Table 4.2 and Figure 4.5, the 1D-Triplet-CNN-online(DeepVOX) method performs the best across all the performance metrics. It is important to note that the performance of all the algorithms is significantly lower in case of cross-noise experiments (Experiments 3 and 5) when compared to the same-noise experiments (Experiments 2 and 4). However, the usage of the proposed DeepVOX features in all the algorithms improves their robustness to the mis-match in the training and testing noise characteristics. Also, the speaker recognition performance in presence of babble noise, compared to the F-16 noise, is observed to be significantly lower. This indicates speech babble as one of the most disruptive speech degradations for speaker recognition tasks [93]. All the algorithms when trained on DeepVOX features, as compared to MFCC, LPC or MFCC-LPC features, gain significant performance improvements

- On an average across the four speaker verification experiments (Experiments 2 to 5) on the degraded Fisher dataset, the usage of DeepVOX features compared to the MFCC-LPC feature, in the 1D-Triplet-CNN-online algorithm improves the verification performance (TMR@FMR=$\{1\%, 10\%$ $\}$) by $\{23.83\%, 10.65\%\}$, reduces the EER by 5.98% and minDCF($P_{tar} = \{0.001, 0.01\}$) by $\{0.007, 0.24\}$. Similarly, for the 1D-Triplet-CNN algorithm, the DeepVOX features improve

speaker verification performance (TMR@FMR=$\{1\%, 10\%\}$) over the MFCC-LPC features by $\{29.69\%, 14.45\%\}$, reduces the EER by 7.11% and minDCF($P_{tar} = \{0.001, 0.01\}$) by $\{0.019, 0.25\}$. For the xVector-PLDA algorithm, the DeepVOX features improve speaker verification performance (TMR@FMR=1%) over the MFCC-LPC features by 8.33%, reduces the minDCF($P_{tar} = \{0.001, 0.01\}$) by 0.002, 0.05. However, a performance (TMR@FMR=10%) loss of 1.52% and an increase in EER by 1.96% were also observed for the xVector-PLDA algorithm using DeepVOX features compared to the MFCC-LPC features. Finally, for the iVector-PLDA algorithm, the DeepVOX features improve speaker verification performance (TMR@FMR=$\{1\%, 10\%\}$) over the best performing baseline feature (MFCC-LPC) by 23.45%, 9.61%. It also reduces the EER by 4.69% and minDCF($P_{tar} = \{0.001, 0.01\}$) by $\{0.008, 0.17\}$. The 1D-Triplet-CNN(DeepVOX) method also outperforms the RawNet2 across all the performance metrics. The TMR@FMR=$\{1\%, 10\%\}$ is increased by 25.32%, 17.62%, EER is reduced by 10.21%, and minDCF($P_{tar} = \{0.001, 0.01\}$) is reduced by 0.004, 0.14. Furthermore, the performance benefits of the proposed method compared to the RawNet2 is even greater in the cross-noise experiments (Experiments 3 and 5), demonstrating its superior resilience to mis-matched degraded audio conditions.

- On an average across the seven speaker verification experiments (Experiments 6 to 12), all the algorithms gain performance benefits when the MFCC, LPC or MFCC-LPC features are replaced with DeepVOX features for training the models. Replacing the best performing baseline features (MFCC-LPC) by DeepVOX features in the 1D-Triplet-CNN-online algorithm improves the verification performance (TMR@FMR=$\{1\%, 10\%\}$) by $14.72\%, 8.7\%$, reduces the EER by 3.67% and minDCF($P_{tar} = \{0.001, 0.01\}$) by $\{0.009, 0.11\}$. Similarly, for the 1D-Triplet-CNN algorithm, the DeepVOX features improve speaker verification performance (TMR@FMR=$\{1\%, 10\%\}$) over the best performing baseline feature (MFCC-LPC) by $8.42\%, 2.46\%$, reduces the EER by 0.99% and minDCF($P_{tar} = \{0.001, 0.01\}$) by $\{0.005, 0.06\}$. For the xVector-PLDA algorithm, the DeepVOX features improve speaker verification performance (TMR@FMR=1%) over the best performing baseline feature (MFCC-LPC) by 2.79% and reduces the minDCF($P_{tar} = \{0.001, 0.01\}$) by $\{0.001, 0.01\}$. However, a performance (TMR@FMR=10%) loss of 2.33% and an increase in

EER of 2.357% were also observed for the xVector-PLDA algorithm using DeepVOX features compared to the MFCC-LPC features. Finally, for the iVector-PLDA algorithm, the DeepVOX features improve speaker verification performance (TMR@FMR=$\{1\%, 10\%\}$) over the best performing baseline feature (MFCC-LPC) by $7.10\%, 22.44\%$. It also reduced the EER by 9.57% and minDCF($P_{tar} = \{0.01\}$) by $\{0.001\}$. However, no significant change in minDCF($P_{tar} = \{0.001\}$) was observed. The 1D-Triplet-CNN(DeepVOX) method also outperforms the RawNet2 across majority of the performance metrics. The TMR@FMR=$\{1\%, 10\%\}$ is increased by $\{5.57\%, 10.6\%\}$, EER is reduced by 3.29%. However, no significant change in minDCF($P_{tar} = \{0.001, 0.01\}$) was observed. Similar to the cross-noise experiments (Experiments 3 and 5) on the degraded Fisher dataset, the 1D-Triplet-CNN(DeepVOX) method vastly outperforms the RawNet2 method in cross-noise experiments (Experiments 11 and 12) on the degraded NIST SRE 2008 dataset.

- In the three speaker verification experiments (Experiments 13 to 15, given in Table 4.4) on multilingual speakers from the NIST SRE 2008 dataset, DeepVOX features perform the best across all the algorithms and metrics, followed by the MFCC-LPC features. The usage of DeepVOX features compared to the MFCC-LPC features, in the 1D-Triplet-CNN-online algorithm, improves the verification performance (TMR@FMR=$\{1\%, 10\%\}$) by $23.95\%, 8.97\%$, reduces the EER by 4.99% and minDCF($P_{tar} = \{0.001, 0.01\}$) by $\{0.02, 0.21\}$. For the 1D-Triplet-CNN algorithm the verification performance (TMR@FMR=$\{1\%, 10\%\}$) improves by $26.81\%, 16.20\%$, the EER reduces by 7.70%, and the minDCF($P_{tar} = \{0.001, 0.01\}$) reduces by $\{0.009, 0.17\}$. For the xVector-PLDA algorithm the verification performance (TMR@FMR=$\{1\%, 10\%\}$) improves by $20.90\%, 10.56\%$, the EER reduces by 5.04%, and the minDCF($P_{tar} = \{0.001, 0.01\}$) reduces by $\{0.005, 0.16\}$. For the iVector-PLDA algorithm the verification performance (TMR@FMR=$\{1\%, 10\%\}$) improves by $18.69\%, 9.33\%$, the EER reduces by 4.92%, and the minDCF($P_{tar} = \{0.001, 0.01\}$) reduces by $\{0.004, 0.12\}$. The 1D-Triplet-CNN(DeepVOX) method also outperforms the RawNet2 across all the performance metrics. The TMR@FMR=$\{1\%, 10\%\}$ is increased by $10.18\%, 5.19\%$, EER is reduced by 3.24%, and minDCF($P_{tar} = \{0.001, 0.01\}$) is reduced by $\{0.019, 0.12\}$.

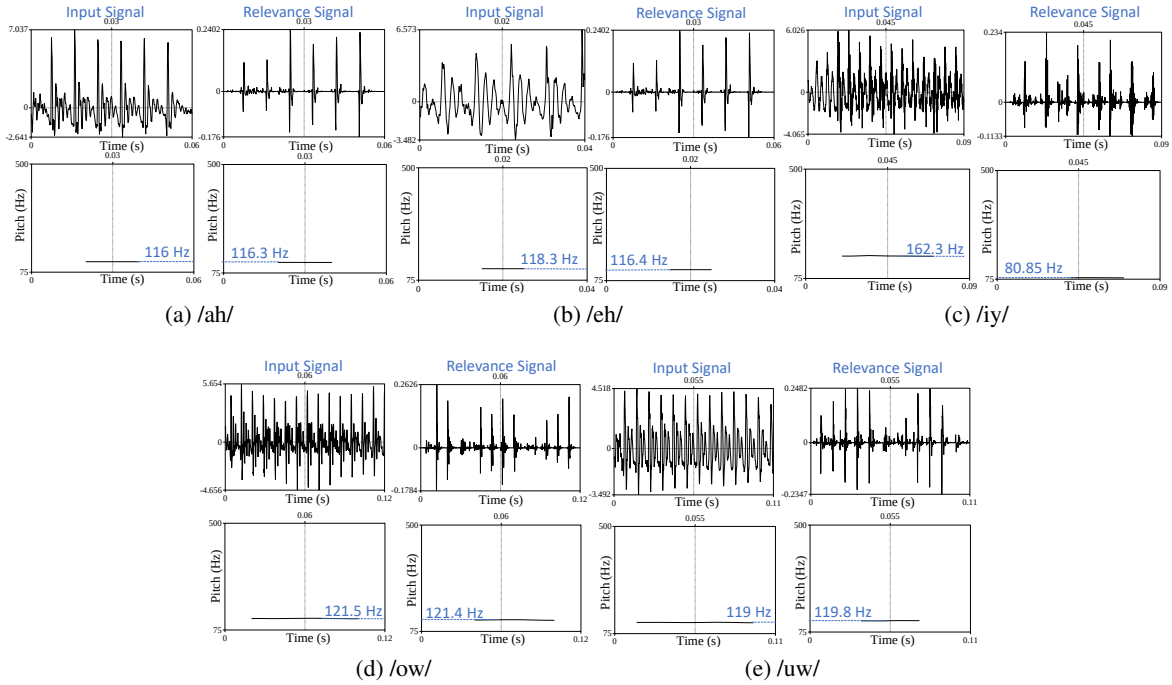
• It is interesting to note the effect of language on verification performance in the Experiments 13 to 15. Best speaker verification performance is achieved in Experiment 13 where the models are trained on English speech data and evaluated on same-language English-only speech audio pairs. However, introduction of same-language multi-lingual speech audio pairs to the evaluation set (in Experiment 14) reduces the verification performance (TMR@FMR=$\{1\%, 10\%\}$) of 1D-Triplet-CNN-online by 3.70% for the DeepVOX features, 14.28% for the MFCC-LPC features, 4.11% for the MFCC features, and 17.46% for the LPC features. Furthermore, re-evaluating the same models on cross-language multi-lingual speech audio pairs in Experiment 15 shows the largest reduction in verification performance, verifying the impact of language-familiarity effect [61,99] in all algorithms and features evaluated in our experiments. It is important to note that the detrimental effects of the language-familiarity effect (in Experiment 14) are observed to be the weakest at 22.49% (performance reduction (TMR@FMR=1%)) for the DeepVOX features compared to 40.76% for the MFCC-LPC features, 39.31% for the MFCC features, and 37.91% for the LPC features using the best-performing 1D-Triplet-CNN-online algorithm.

• Additionally, as given in Table 1, we also investigate language-based demographic bias in the different speaker verification methods proposed and used in this work. In this experiment, we divide the evaluation trials based on the language spoken in them. For example, for evaluating speaker verification performance on a given language, all the genuine trials only consist of speech in that particular language. However, the impostor trials may include other languages as well. This experiment aims to understand the presence and effect of demographic biases in speaker verification methods trained only using the English language and evaluated on a wide variety of non-English languages. Therefore, all the models are trained using English-only speech and evaluated on non-English speech trials. In these experiments, we notice that the average speaker verification performance across all the methods (TMR@FMR=1%) varies from 40.48% for language ID 'CFR' to 91.58% for language ID 'WUU'. This demonstrates the presence of demographic bias in speaker recognition methods trained using limited speech data from a single language.

• In the experimental results given in Table 4.5 and illustrated in Figure 4.6, we notice a gradual decrease in verification performance (across all algorithms and features) with the decrease in length of audio samples in the testing data. However, the loss in performance is observed to be much lower with the usage of DeepVOX features compared to MFCC, LPC, or MFCC-LPC features across all the algorithms. The 1D-Triplet-CNN-online algorithm using DeepVOX features sufferes a performance (TMR@FMR=10%) reduction of 10%, compared to a reduction of 32% using MFCC-LPC features, 45% using MFCC features, 34% using LPC features, when the audio length is reduced from 3.5 seconds to $0.5seconds$. Similar trends were observed for the 1D-Triplet-CNN algorithm where a performance loss of 11%, 18%, 25%, and 25% is observed for the DeepVOX, MFCC-LPC, MFCC, and LPC features respectively. For the xVector-PLDA algorithm a performance loss of 47%, 61%, 59%, and 54% is observed for the DeepVOX, MFCC-LPC, MFCC, and LPC features respectively. For the iVector-PLDA algorithm a performance loss of 40%, 54%, 55%, and 55% is observed for the DeepVOX, MFCC-LPC, MFCC, and LPC features respectively. Finally, for the RawNet2 algorithm a performance loss of 48% is observed when the length of raw input audio is reduced from 3.5 seconds to $0.5seconds$. It is important to note that, compared to the 1D-Triplet-CNN based algorithms, relatively larger performance losses are observed for the iVector-PLDA, xVector-PLDA, and RawNet2 algorithms, across all the features. However, using the DeepVOX features improves the robustness of even the iVector-PLDA and xVector-PLDA algorithms when performing speaker verification on speech samples of limited duration, thereby, asserting the effectiveness of the DeepVOX features in the task.

4.5 Ablation Study of DeepVOX

In the previous section, we discussed the performance benefits of the proposed DeepVOX features using different algorithms, multiple datasets, and a number of different experimental protocols. In this section, similar to [117], we attempt to analyze the type of speech information being extracted and encoded by the 40-dimensional DeepVOX features using a technique called 'Guided Backpropagation' [157]. Such an analysis will help us understand the components of a speech



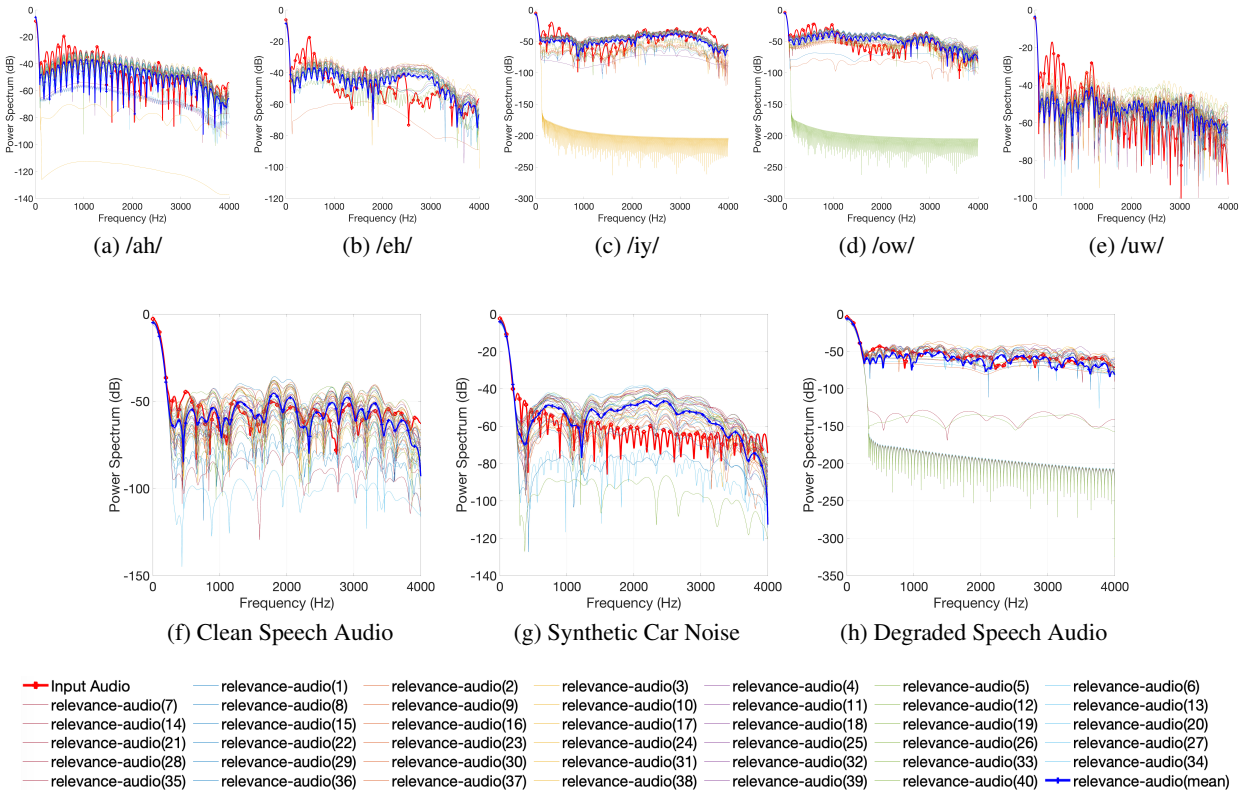
(a) /ah/ (b) /eh/ (c) /iy/

(d) /ow/ (e) /uw/

Figure 4.7: A visual comparison of the waveforms and F0 contours for five different phonemes (/ah/,/eh/,/iy/,/ow/,and /uw/) and their corresponding relevance signals obtained for the proposed DeepVOX model, using the Praat [27] toolkit. Each sub-figure shows: the input signal (top-left), the relevance signal (top-right), F0-contour plot for input signal (bottom left), and F0-contour plot for relevance signal (bottom-right).

audio that are deemed important, by the DeepVOX model, in the context of speaker recognition.

In this analysis, we use the DeepVOX model trained for Experiment #1 on the VOXCeleb2 dataset, due to the large number of training speakers and a wide variety of audio recording conditions in the training data. For evaluation, we choose audio samples from the TIMIT [60] dataset due to the availability of ground-truth information for analysis of frequency sub-bands essential for speaker recognition in the TIMIT dataset [62, 90, 125]. For analysing the DeepVOX method, we feed an input audio sample to the trained DeepVOX model and extract the 40-dimensional DeepVOX features. Guided backpropagation is then used individually on each of the 40 features to estimate the corresponding relevance signals. The relevance signal in this case refers to the portion of input audio signal (in the frequency domain) that the DeepVOX model fixates on to extract a corresponding DeepVOX feature. The 40 relevance signals corresponding to the 40 DeepVOX features are aggregated to estimate the mean relevance signal. The mean relevance signal is then



(a) /ah/ (b) /eh/ (c) /iy/ (d) /ow/ (e) /uw/

(f) Clean Speech Audio (g) Synthetic Car Noise (h) Degraded Speech Audio

Figure 4.8: Power Spectral Density(PSD) plots for the analysing the representation capability of the learned DeepVOX filterbank on a variety of speech audio samples from TIMIT dataset and synthetic noise audio samples from NOISEX-92 dataset.

analysed, as given below, to characterize the properties of the speech signal extracted by the DeepVOX features important for performing speaker recognition:

Fundamental Frequency (F0) Extraction by the DeepVOX: In this experiment, illustrated in Figure 4.7, we extract speech utterances corresponding to the five phonemes /ah/, /eh/, /iy/, /ow/, /uw/ from a randomly chosen speaker in the TIMIT dataset. The speech audio of these phonemes is then fed to the trained DeepVOX model to extract corresponding DeepVOX features. Guided backpropagation is then used to extract the corresponding relevance signals. The input speech signal and the corresponding mean relevance signal are then compared using the Praat [27] toolkit, as illustrated in Figure 4.7. While the waveform representation of the original input signals and the corresponding mean relevance signals differ visually, pitch contour analysis of the signals reveals that the relevance signal successfully captures the F0 contours of the input speech signal for the

majority of the phonemes. This indicates that the DeepVOX architecture successfully extracts and uses fundamental frequency (F0) (a vocal source feature), for representing the human voice. This could be seen as a direct effect of the presence of phase information in the raw input speech audio, as phase information in speech audio captures rich vocal source information [86].

Operational Frequency-range of the DeepVOX Model: Similar to [117], we represent the input audio signal and corresponding relevance signals on the Power Spectral Density (PSD) plots (given in Figure 4.8 [(a) to (e)]). The PSD plots are inspected for portions of frequency bands where the input audio signal (given by red color) and the corresponding mean audio signal (given by blue color) are overlapping in Figure 4.8. This is done to compare and identify the frequency components of the input audio signal that are reliably captured by the DeepVOX (in the relevance signal) and are essential for performing speaker recognition. The 40 relevance signals corresponding to the 40 DeepVOX features that constitute the mean relevance signal are also shown on the Power Spectral Density (PSD) plots.

The trained DeepVOX model is observed (in Figure 4.8 [(a) to (e)])) to reliably model the input speech signal in the frequency range of 0 to 4000Hz. However, a better modeling performance is observed in the mid/high-frequency range of 2000Hz to 4000Hz, which is known to contain more discriminative information in the context of speaker recognition in the TIMIT dataset [62, 90, 125]. An informal listening test of the relevance signals extracted by the DeepVOX model lends to intelligible reproduction of input speech audio. This confirms that the DeepVOX model can use spectral information from a large frequency range (0 to 4000Hz) for performing speaker recognition.

Effect of Audio Degradation on the DeepVOX: As shown in Figure 4.8 [(f) to (h)], we also compared the response of the trained DeepVOX model on a degraded audio sample, the constituent clean speech sample from the TIMIT [60] dataset, and the additive synthetic car noise from the NOISEX-92 dataset [166]. This is done to analyze the robustness of the DeepVOX model to audio degradations. The DeepVOX model is observed to model the speech in both the clean and

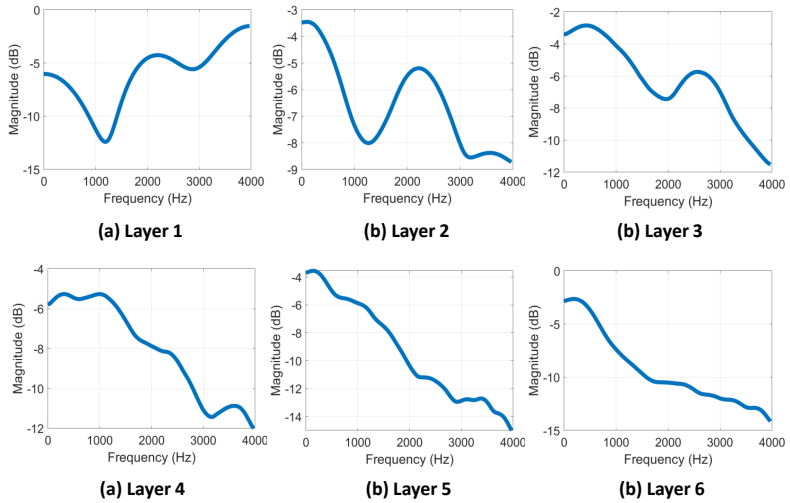


Figure 4.9: Cumulative layer-wise magnitude frequency response of the DeepVOX model trained on the VoxCeleb2 dataset

degraded speech audio reliably while failing to model the noise in the synthetic car noise sample. This demonstrates the ability of the DeepVOX network to selectively model the speech audio and reject the background noise in an audio sample for performing speaker recognition.

Layer-wise magnitude frequency response of the DeepVOX: Finally, we also plotted (see Figure 4.9) the layer-wise cumulative magnitude frequency response of the convolution filters in the DeepVOX model trained on the VoxCeleb2 dataset. Here we observed that while the initial three layers behave as a multi-band pass filter, the later layers act as low-pass filters. Specifically, the first three layers' cumulative magnitude frequency response shows peaks in the frequency range of 0-800Hz and 1500-3000HZ. Comparing to the acoustic characteristics of the human voice in American English [73], the first peak (0-800Hz) is specifically suited for capturing the fundamental frequency (F0) and first formant (F1) of the human voice (the average F0 is 195Hz and average F1 is 595Hz) and the second peak (1500-3000HZ) can capture the second (F2) and third (F3) formants of the human voice (the average F2 is 1734Hz and the average F3 is 2826Hz). Therefore, the initial layers of the DeepVOX model learn to capture important speaker-dependent speech characteristics (F0, F1, F2, and F3) from input speech audio and are well-suited for application in a speaker recognition system.

4.6 Conclusion

The performance of short-term speech feature extraction techniques, such as MFCC, is dependent on the design of filterbanks, driven by psychoacoustic studies involving human hearing and perception [176]. Mel-Frequency bank and Gammatone-frequency bank are two such examples of handcrafted filterbanks used in MFCC and GFCC features, respectively. While such feature extraction techniques are easy to use and do not require any training data, they do not adapt well to the changes in the speech audio quality owing to degradations such as background noise, channel distortion, etc. Therefore, it is beneficial to develop feature extraction techniques, such as the proposed DeepVOX algorithm, that can adapt to target speech characteristics and is robust across different types of audio degradations, as evident in the experimental results. The proposed technique improves speaker recognition performance vastly across almost all the experiments. The frequency analysis of the learned DeepVOX filterbanks indicates that the proposed model can extract spectral information from a large frequency range (0 to 4000Hz) and also extract the fundamental frequency (F0) information for representing the speaker in speech audio. It is also important to make note of cases such as Experiment 8 in Table 4.3, where certain combinations of noise characteristics in the training and testing sets create challenging scenarios where the proposed DeepVOX feature does not outperform the baselines. Therefore, it is important to continue research in the further development of feature extraction algorithms that build upon the currently proposed algorithm and further improve the speaker verification performance in extensively challenging scenarios. As discussed in section 4.2.1.1, the proposed DeepVOX algorithm has a limitation of only training on 200 audio frames at a time, hence it cannot benefit from training on longer audio samples in the training set. We plan to extend our DeepVOX model by incorporating methods for automatically learning from audio samples of varying lengths, as seen in methods that use Recurrent Neural Networks (RNN) for speech processing.

CHAPTER 5

VOCAL STYLE ENCODING FOR SPEAKER RECOGNITION AND SPEECH SYNTHESIS

Portions of this chapter appeared in the following publication:

Chowdhury, Anurag, Ross, Arun, and Prabu David. "DeepTalk: Vocal Style Encoding for Speaker Recognition and Speech Synthesis." IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (2021).

5.1 Introduction

In the previous chapters, we developed a 1D-CNN-based method, called DeepVOX, to automatically discover features directly from raw speech audio for performing speaker recognition in degraded audio signals. Furthermore, DeepVOX was shown to successfully encode physiological speech characteristics, such as vocal tract and vocal source features, for extracting highly discriminative speaker-dependent speech characteristics. This chapter will introduce a method, called DeepTalk, for extracting behavioral speech features, such as vocal style and prosody for performing speaker recognition in non-ideal audio conditions. Furthermore, we will also combine the DeepTalk-based behavioral speech features with several state-of-the-art physiological speech feature-based speaker recognition methods for improving speaker recognition performance in non-ideal audio conditions.

Speaker recognition is the task of determining a person's identity from their voice. The human voice as a biometric modality is a combination of physiological and behavioral characteristics. The physical traits of the voice production system determine the human voice's physiological characteristics [42], while the prosodic (pitch, timbre) and high-level (lexicon) traits impart the human voice's behavioral characteristics. Most automatic speaker recognition systems use only the

physiological speech features due to their high discriminability and ease of characterization [120]. However, such automatic speaker recognition systems are vulnerable to audio degradations, such as background noise and channel effects [66]. Behavioral speech characteristics, while being vulnerable to intra-user variations, are considered robust to audio degradations [35]. Behavioral speech features also complement the speaker-dependent speech characteristics captured by physiological speech features, and can be combined to improve the speaker recognition performance [19]. Behavioral speech features, when used judiciously, can help in the development of robust speaker recognition systems.

A person's behavioral speech characteristics are defined by their long-term and short-term speaking habits, referred to as their 'vocal style.' Long-term vocal styles are acquired over time and are influenced by social environments and native language [19]. Short-term vocal styles are more volatile and are influenced by the target audience (addressing a crowd versus talking over the phone) and emotional state [171]. Furthermore, a apart from a speaker's idiolect, their vocal anatomy also influences their behavioral speech characteristics, thus constraining the differences between a their physiological and behavioral speech features [144]. Behavioral speech characteristics have been used for performing speaker recognition [151]. Most of these techniques require speech data annotated at the word- and frame-level for extracting behavioral speech features [151], posing a challenge for their development. Recently, deep learning-based methods have been developed to learn vocal style features from speech data without any word- or frame-level annotations [81, 150, 168]. However, most of these methods use handcrafted speech features, such as Mel-frequency Cepstral Coefficients (MFCC) [120], to represent the input audio, thus contending with the vulnerabilities and performance bottlenecks of handcrafted features [41].

In this work, we develop a speech encoder called DeepTalk, to capture behavioral speech characteristics directly from raw speech audio without any word- or frame-level annotations. DeepTalk's speaker recognition performance is evaluated on multiple challenging datasets. DeepTalk is further combined with physiological speech feature-based speaker recognition methods to improve state-of-the-art speaker recognition performance in challenging audio conditions. The fidelity of

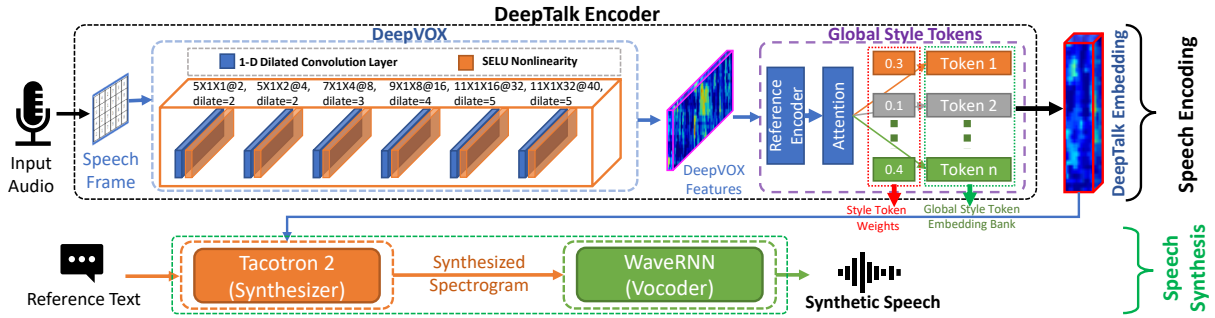


Figure 5.1: The speech encoding and speech synthesis branches of the proposed DeepTalk architecture.

DeepTalk-based vocal style features is evaluated by integrating it with a Tacotron2-based TTS synthesizer [150] to generate synthetic speech audios. The Deeptalk-based synthetic speech audios are shown to be indistinguishable from real speech audios in the context of speaker recognition. Therefore, DeepTalk serves the dual purpose of improving both speaker recognition and speech synthesis performance.

5.2 DeepTalk

The DeepTalk architecture (Fig. 7.1) consists of separate speech encoding and speech synthesis branches, as follows:

5.2.1 Speech Encoding

The speech encoding branch feeds a raw input audio into a DeepVOX [41] network to extract short-term speech features, called DeepVOX features. DeepVOX is a 1D-CNN based speech filterbank that extracts speaker-dependent speech features directly from raw speech audio. DeepVOX features are then fed to a Global Style Token (GST)-based [168] prosody embedding network to extract the DeepTalk embedding. The GST network uses a 2-dimentional Convolution Neural Network (2D-CNN) followed by a single-layer 128-unit unidirectional Gated Recurrent Unit (GRU) to extract a fixed dimensional reference encoding from the DeepVOX features. The reference encoding is then passed to a bank of ten randomly-initialized 128-dimensional style token embeddings called the Style Token Layer [168]. The style token embeddings serve as basis vectors of a Style Token Space representing the different vocal styles in the training data. Finally, an attention

module is used to represent the reference encoding as a weighted combination of different style tokens embeddings, namely the DeepTalk embedding. We train both the DeepVOX and GST networks together using a triplet-based speaker embedding learning framework [41] to maximize the speaker-dependent vocal style information in the DeepTalk embedding. This allows DeepVOX to learn the speech representation best-suited for vocal style extraction using the GST network.

5.2.2 Speech Synthesis

The speech synthesis branch feeds the DeepTalk embedding and a reference text into a Tacotron2-based synthesizer to generate a Mel spectrogram, which is then converted to the synthetic speech waveform using a WaveRNN-based neural vocoder [85]. The synthetic speech's similarity to the target speaker's original speech is then qualitatively and quantitatively evaluated. The qualitative evaluation is done by manually listening to the synthetic speech and visually comparing the spectrograms of the original and synthetic speech (see Fig. 5.2). The quantitative evaluation is done by comparing the two audios using speech embedding techniques (see Fig. 7.4). For training the Tacotron2 model, we use a two-phase approach. In the first phase, we train the Tacotron2 and WaveRNN models on speech audio from a large group of speakers in the VoxCeleb2 [43] dataset to learn general voice characteristics present in VoxCeleb2. In the second stage, we finetune the trained models on a small set of speech samples (30 minutes long) of a target speaker to enable high fidelity vocal style transfer from the target speech to the synthetic speech.

5.3 Dataset and Experiments

In this section, we discuss the datasets, experimental protocols, and baseline methods used to evaluate and compare DeepTalk's speaker recognition performance. Speech data used in this work have been sampled at 8 kHz.

5.3.1 Datasets

Table 5.1: Speaker verification results using the iVector-PLDA **[M1]**, xVector-PLDA **[M2]**, 1D-Triplet-CNN **[M3]**, DeepVOX **[M4]**, DeepTalk **[M5]**, and 1D-Triplet-CNN (DeepVOX) + DeepTalk **[M6]** methods. Here, **P1** = VoxCeleb2, **P2** = NIST SRE 2008, **P3** = Degraded NIST SRE 2008 (Babble), and **P4** = Degraded NIST SRE 2008 (F16).

Exp. #	Models	Train set/Test set	TMR@FMR=1%	minDCF	EER (in %)	Exp. #	Models	Train set/Test set	TMR@FMR=1%	minDCF	EER (in %)
1	M1	P1/P1	86.16	2.04	5.39	19	M4	P1/P1	**91.98**	**1.47**	**2.91**
2		P2/P2	48.7	5.68	12.37	20		P2/P2	81.05	2.85	4.45
3		P3/P3	39.57	6.37	13.53	21		P3/P3	70.16	3.51	7.44
4		P4/P4	22.73	8.5	21.13	22		P4/P4	62.4	4.21	7.25
5		P3/P4	6.03	9.93	35.24	23		P3/P4	15.46	9.25	22.46
6		P4/P3	9.5	9.59	34.11	24		P4/P3	**35.05**	**7.16**	**15.22**
7	M2	P1/P1	55.75	5.03	11.25	25	M5	P1/P1	87.58	2.09	4.96
8		P2/P2	24.2	8.01	14.15	26		P2/P2	66.73	3.52	4.44
9		P3/P3	22.44	8.35	15.24	27		P3/P3	50.7	4.47	6.7
10		P4/P4	17.15	9.01	20.88	28		P4/P4	61.53	4.27	6.53
11		P3/P4	7.71	9.73	34.95	29		P3/P4	10.69	9.76	28.45
12		P4/P3	12.17	9.56	27.54	30		P4/P3	7.54	9.88	34.04
13	M3	P1/P1	82.09	2.65	5.42	31	M6	P1/P1	91.69	1.52	3.14
14		P2/P2	52.5	5.2	8.18	32		P2/P2	**83.56**	**2.54**	**3.91**
15		P3/P3	35.25	6.54	11.4	33		P3/P3	**76.86**	**3.23**	**6.14**
16		P4/P4	38.50	7.08	14.96	34		P4/P4	**66.52**	**3.6**	**5.92**
17		P3/P4	8.83	9.84	29.49	35		P3/P4	**17.36**	**9.15**	**21.49**
18		P4/P3	20.00	8.85	22.64	36		P4/P3	29.37	7.73	18.09

Table 5.2: Speaker verification results on synthetic audio samples in presence of 1211 background speakers from the VOXCeleb1 dataset.

Exp. #	Models	Evaluation Condition	TMR@FMR=0.1%	TMR@FMR=1%	minDCF	EER (in %)
1	1D-Triplet-CNN (MFCC-LPC)	Real - Real	75	97.5	0.93	1.46
2		Real - Synthetic (Baseline)	5	42	6.53	22.00
3		Real - Synthetic (DeepTalk)	7	46	4.33	6.38
4	1D-Triplet-CNN (DeepVOX)	Real - Real	100	100	0	0
5		Real - Synthetic (Baseline)	11	44	6.36	13.61
6		Real - Synthetic (DeepTalk)	86	100	0.50	0.61
7	DeepTalk	Real - Real	82.5	100	0.076	0.07
8		Real - Synthetic (Baseline)	0	7	8.46	10.23
9		Real - Synthetic (DeepTalk)	50	100	0.17	0.23

- **VoxCeleb2:** We use the VoxCeleb2 [43] dataset to perform speaker recognition experiments on speech collected in unconstrained scenarios. We use speech extracted from one randomly chosen video for each of the $5,994$ celebrities in the training set and the 118 celebrities in the test set. Speech samples longer than 5 seconds are split into multiple non-overlapping 5 second long speech samples.

- **NIST SRE 2008:** We use the NIST SRE 2008 [1] dataset to perform speaker recognition experiments on multilingual speech data, captured under varying ambient conditions and channel effects. Additionally, we degrade the NIST SRE 2008 dataset with F-16 and Babble noise from the NOISEX-92 dataset [165], to increase the difficulty of the task. For our experiments, we choose speech data from the 'phonecall' and 'interview' speech types collected under audio conditions labeled as '10-sec', 'long', and 'short2' across 1336 speakers. Speech data from a random subset of 200 speakers is reserved to evaluate the models, while the rest is used for training.

5.3.2 Speaker Recognition Experiments

We perform multiple experiments (Table 6.3) to evaluate and compare the speaker recognition performance of DeepTalk-based behavioral speech features with several baseline speaker recognition methods. The iVector-PLDA [50] and the xVector [154] algorithms are used as our first and second baseline methods, respectively, due to their robustness to channel variabilities. The MSR Identity Toolkit [143] is used to perform the iVector-PLDA [50] experiments in this work. The PyTorch-based implementation [42] of the xVector [154] algorithm paired with a gPLDA-based matcher [143] is used to perform the xVector-PLDA-based experiments in this work. The 1D-Triplet-CNN [42] algorithm is used as our third baseline, as it can extract both speech production- and speech perception-based physiological speech features. The DeepVOX [41] algorithm is used as our fourth baseline, as it can extract vocal source- and vocal tract-based physiological speech features. Finally, the DeepTalk and DeepVox methods are combined at a weighted score level, in a 1:3 ratio (chosen empirically), to evaluate the speaker recognition benefits of combining physiological and behavioral speech features.

5.3.3 Speaker Recognition Results

We report the speaker verification performance (see Table 6.3) using True Match Rate at a False Match Rate of 1% (TMR@FMR=1%), minimum Detection Cost Function (minDCF) and Equal Error Rate (EER in $\%$). The minDCF is computed at a prior probability of 0.01 for the specified target speaker (P_{tar}) with a missed detection cost of 10 (C_{miss}).

- In experiments 1, 7, 13, 19, 25, and 31, the VoxCeleb2 dataset is used to perform the speaker recognition experiments on a large number of speech audios collected in unconstrained scenarios. Here, the DeepVOX method and its score level fusion with the DeepTalk method obtain comparable performance and outperform all the other methods.
- In experiments 2, 8, 14, 20, 26, and 32, the NIST SRE 2008 dataset is used to perform the speaker recognition experiments on multi-lingual speech audios portraying challenging real-life

audio conditions. Here, the score level fusion of the DeepTalk and DeepVOX methods performs the best, demonstrating its robustness to challenging audio conditions.

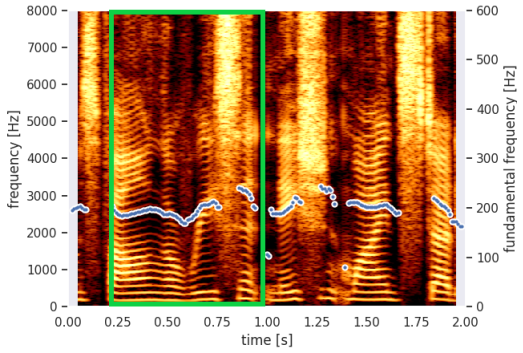
• In experiments 3-6, 9-11, 15-18, 21-24, 27-30, and 33-35, speaker recognition experiments are performed on the degraded NIST SRE 2008 dataset. Here, the score level fusion of the DeepTalk and DeepVOX methods performs the best, validating its robustness to audio degradations.

• Overall, the DeepTalk method outperforms all but the DeepVOX-based speaker recognition algorithm. This demonstrates the highly discriminative characteristics of the behavioral speech features extracted by DeepTalk method. The score level fusion of the DeepTalk and the DeepVOX methods further improves the speaker recognition performance across majority of the experiments. This establishes the performance benefits of combining the physiological (in DeepVOX) and behavioral (in DeepTalk) speech characteristics. We also performed score level fusion of the DeepTalk method with 1D-Triplet-CNN, xVector-PLDA and iVector-PLDA-based methods. Similar performance improvements were noted across majority of the experiments, with the best results achieved by the fusion of DeepTalk with DeepVOX, followed by 1D-Triplet-CNN, iVector-PLDA, and xVector-PLDA.

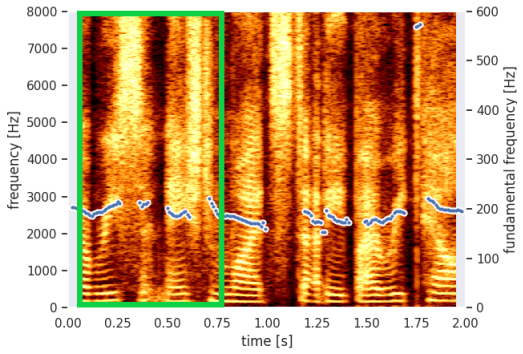
5.3.4 Speech Synthesis Experiments and Results

We performed multiple speech synthesis experiments, listed below, to demonstrate and analyse DeepTalk's vocal style encoding ability (Fig. 5.2 and 7.4). In these experiments, speech audio from the VOXCeleb2 dataset is used to train DeepTalk. The trained models were then adapted to high-quality speech audio from four different speakers (two male and two female) from the Librispeech dataset [126] as well as internal sources. DeepTalk's speech synthesis performance was also compared to a baseline Tacotron2-based speech synthesis method [82]. The generated synthetic audio samples can be reviewed here.

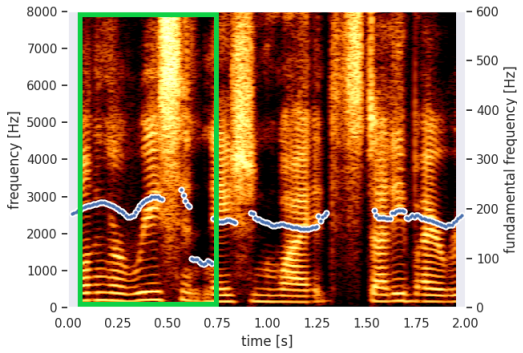
• Copy synthesis experiment: Here, the DeepTalk method extracts speech characteristics from an input audio and combines it with the text transcript (of the same input sample) to recreate



(a) Original speech



(b) Baseline Tacotron2



(c) DeepTalk

Figure 5.2: Spectrogram representation (overlaid with F0 contour) of a speech sample from a sample speaker and its corresponding synthetic speech samples generated using the baseline Tacotron2 model and the DeepTalk model, respectively. The green overlay boxes indicate the locations of corresponding speech segments across the three spectrograms.

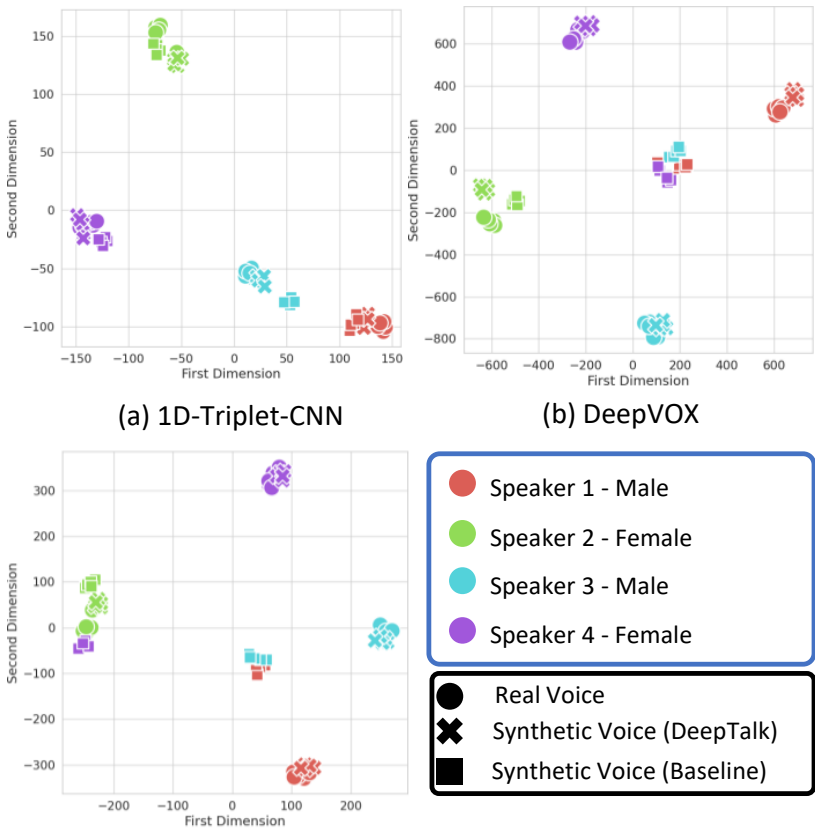


Figure 5.3: t-SNE plots of the speech embeddings of real and synthetic voice samples of four different speakers, extracted by three different speech encoders. DeepTalk's synthetic speech is embedded much closer to the real speech by all the speech encoders, as compared to the baseline synthetic speech.

the input audio. The spectrogram representation of DeepTalk's synthetic speech displays greater visual similarity to the original speech, especially at frequencies higher than 2500Hz compared to the baseline (Fig. 5.2). Furthermore, the high visual similarity between the F0 contours of the original speech and DeepTalk's synthetic speech (indicated by green overlay boxes in Fig. 5.2) demonstrates DeepTalk's efficacy at vocal style modeling [103].

- Speaker Matching Experiment: Here, the 1D-Triplet-CNN, DeepVOX, and DeepTalk-based speech encoding methods extract speech embeddings from original and synthetic (both DeepTalk and baseline) speech samples for the four different speakers. The speech embeddings are then visualized using t-SNE [100](Fig. 7.4). All the speech samples used in this experiment contain different speech utterances, ensuring a text-independent speaker matching scenario. Across all the

speech encoding methods, the speech samples synthesized by the DeepTalk method are embedded much closer (mean euclidean distance 45) to the corresponding real voice samples from the same speaker, when compared to the baseline method (mean euclidean distance 189). This demonstrates DeepTalk's ability to generate near-indistinguishable synthetic speech samples in the context of speaker recognition.

Furthermore, we also perform speaker recognition experiments using 1D-Triplet-CNN, DeepVOX, and DeepTalk-based speaker verification methods on the synthetic audio samples of the four different speakers generated using DeepTalk and baseline speech synthesis frameworks. Here, we also include speech data from 1211 speakers in the VoxCeleb1 dataset to serve as background speakers. In these experiments, we perform speaker verification under the following evaluation conditions:

– Real-Real: In this experiment, real speech samples from a subject are compared to their real speech samples to establish a speaker verification baseline performance on real speech evaluation trails.
– Real-Synthetic (Baseline): In this experiment, real speech samples from a subject are compared to their synthetic speech samples generated using the baseline speech synthesis framework.
– Real-Synthetic (DeepTalk): In this experiment, real speech samples from a subject are compared to their synthetic speech samples generated using the DeepTalk-based speech synthesis framework.

Across all the speaker verification methods, the best performance is obtained on the Real-Real evaluation condition followed by Real-Synthetic (DeepTalk) and Real-Synthetic (Baseline) evaluation conditions. This reinforces DeepTalk's superior ability at generating realistic synthesis speech audio in the context of speaker recognition.

5.4 Ethical Implications

In this work, we demonstrate DeepTalk's ability to reliably model the vocal style of a given speaker and transfer it to a synthetic speech with high fidelity. While this technique can improve the user-experience of Speech Generating Devices (SGD) [140] and digital voice assistants, several

concerns are raised by its potential misuse for creating DeepFake speech. For example, in the past, DeepFake speech has been used to mimic an influential person's voice for defrauding [159]. Therefore, such a technology should be used responsibly while adhering to appropriate privacy-protection laws.

5.5 Conclusion

Behavioral speech characteristics are robust to audio degradations and complement physiological speech characteristics' biometric utility. Therefore, it is beneficial to develop vocal style modeling techniques, such as the proposed DeepTalk algorithm and combine it with physiological speech features for improving speaker recognition performance, as evident in the experimental results. DeepTalk has also been integrated with a Tacotron2-based TTS synthesizer to generate highly-realistic synthetic speech, demonstrating its efficacy at high-fidelity vocal style modeling. Therefore, it is essential to continue developing vocal style modeling algorithms and combine them with physiological speech characteristics to improve speaker verification and speech synthesis performance in challenging audio conditions.

CHAPTER 6

THE EFFECT OF VOCAL STYLE VARIATION IN SPEAKING VS SINGING VOICE ON SPEAKER RECOGNITION

Portions of this chapter appeared in the following publication:

Chowdhury, Anurag, Cozzo, Austin, and Arun Ross. "JukeBox: A Multilingual Singer Recognition Dataset" INTERSPEECH (2020).

6.1 Introduction

In the previous chapters, we focused on developing speaker embedding methods for extracting robust speaker-dependent speech characteristics, both physiological and behavioral, for improving speaker recognition performance in non-ideal audio conditions. This chapter will study the problem of speaking style variability across spoken and singing voices and its effect on speaker recognition. Towards that end, we will introduce an annotated multi-lingual singing voice dataset, called JukeBox, for facilitating the development and evaluation of speaker recognition methods on the unique challenges of the singing voice. Specifically, some of the key challenges of performing speaker recognition in the singing voice include increased intra-user variance due to increased vocal range of the singing voice and a wide variety of background noise such as background chorus and instrumentation.

Speaker recognition entails comparing two audio samples encompassing human voice and determining if the voices pertain to the same individual. A majority of speaker recognition research has focused on modeling the speaker-dependent characteristics from conversational or spoken voice data [91]. However, the spoken voice only exhibits a limited range of possible speaker dynamics [163]. As a result, such speaker recognition systems generalize poorly to a wide variety of speaking styles and vocal effort [152]. The singing voice is one such example of a speaking style [112], where the speaker-dependent voice characteristics depart heavily from the spoken

voice of the same speaker. Apart from the perceived differences in intensity, pitch, and timbre, there are also differences in the physiological formation of sung speech [45], especially when considering a trained singer [31]. The different styles of singing further diversify the acoustic differences between spoken and singing speech [158], leading to several challenges for speaker recognition systems. One of the primary challenges of speaker recognition from singing is the increased intra-user variance and decreased inter-user variance due to intentional voice modulation, across a broad acoustic spectrum [163]. In addition, the presence of background music and chorus increases the challenges of the task. Thus, a speaker recognition system's ability to correctly match a singer's voice across multiple songs can be used to assess its robustness.

However, there appears to be limited amount of work done on this topic. Some of the relevant early literature treat singing voice as a speaking style and cluster it using speaker clustering algorithms [111, 112]. In another work [129], the authors use singing voice to perform speaker recognition; however, no cross-modal experiments were done, i.e. training a model on speaking data and testing on singing data (or vice versa). This work was extended in [37] to evaluate cross-modal speaker recognition; however, poor performance was reported. Notably, the datasets used in [37, 111, 112, 129] were limited to a small set (≤ 50) of speakers.

One key reason behind the underrepresented research focus on speaker recognition from singing voice, i.e., singer recognition, is the lack of sufficient development and evaluation data. A review of currently existing music datasets for research (in Table 6.1) reveals two relevant datasets: the Million Song Dataset (MSD) [26] and the Free Music Archive (FMA) [47]. MSD contains $1,000,000$ songs from $44,745$ artists/groups. However, the data is available only in the form of audio features and not raw audio, which forces a speaker recognition algorithm to work with a predetermined feature-set. FMA, on the other hand, contains $106,574$ songs from $16,341$ artists/groups. Here, the 'artist/group' label refers to the associated music group/band and not necessarily the individual singer, who might change over time. For example, both Ozzy Osbourne and Ronnie James Dio have sung songs under the artist label of Black Sabbath, thus making group/band labels unsuitable for training or testing a speaker recognition system.

Table 6.1: A list of related music datasets compared to the *JukeBox* dataset.

Dataset	Number of Samples	Number of Artists	Label	Raw Audio
UT-Sing [111]	165	33	Singer	Yes
MusiClef [146]	1,355	218	Artist / Group	No
Homburg [76]	1,886	1,463	Artist / Group	Yes
1517-Artists [148]	3,180	1,517	Artist / Group	Yes
Unique [149]	3,115	3,115	Artist / Group	Yes
USPOP [25]	8,752	400	Artist / Group	No
CAL10K [164]	10,271	4,597	Artist / Group	No
MagnaTagATune [94]	16,389	270	Artist / Group	Yes
Codiach [109]	20,849	1,941	Artist / Group	No
FMA [48]	106,574	16,341	Artist / Group	Yes
OMRAS2 [106]	152,410	6,983	Artist / Group	No
MSD [26]	1,000,000	44,745	Artist / Group	No
JukeBox	**7,000**	**936**	**Singer**	**Yes**

Therefore, in this work, we assemble *JukeBox*, a singing voice dataset annotated with singer, gender, and language labels for the development and evaluation of speaker recognition methods. In the next few sections, we will describe in detail this dataset, the data collection procedure, several experimental protocols, and analyze the performance of state-of-the-art speaker recognition methods on the dataset.

6.2 *JukeBox* Dataset

The *JukeBox* dataset contains 467 hours of singing audio data sampled at 16 KHz, downloaded from the Internet Archive (IA) [11]. There is a total of 936 different singers in the dataset, of which 533 are male. Figures 6.1 and 6.2 summarize the different languages and the distribution of the length of songs in the *JukeBox* dataset. The songs in the *JukeBox* dataset:

- are sung in 18 different languages, as shown in Figure 6.1, where almost one-fifth of the singers in the dataset sing in non-English languages (i.e., a language other than English).
- are recorded under a wide variety of acoustic environments and recording apparatus, ranging from highly-constrained studio recording setups to completely-unconstrained live concerts.

• contain multiple singers apart from the person-of-interest (POI), for example, vocal duets with overlapped singing and background chorus.

• contain different types of background music (such as drums, piano, or other instrumentation), thus adding to the difficulty of performing speaker recognition.

6.2.1 Data collection procedure

The *JukeBox* dataset was assembled as follows.

• **Candidate list creation for artists of interest:** We started by compiling a list of artists from Wikipedia, who were tagged as “singer". This yielded a list of $5,046$ artists of interest (AOI) from a variety of languages and genres (such as Pop, R&B, Rock, Jazz, Folk, Classical, etc.), with associated metadata such as country of origin (~ 18 different countries) and years active.

• **Candidate list creation for songs of interest:** The candidate list for AOI was used to query Spotify’s song database [16] to generate a list of $162,311$ songs. This list was then cross-referenced against IA’s repository to generate a list of downloadable songs of interest (SOI). We chose IA as our audio source due to its (a) large collection of audio, (b) public accessibility, (c) nearly unrestricted download access [12], and (d) re-distribution permission for non-commercial purposes.

• **Downloading songs of interest:** The IA repository often contains multiple copies of a song, differing in their audio duration, recording conditions (such as studio versus live versions), and singers (such as original versus cover artists). We specifically avoided cover artists to remove multiple versions of a song and ensure the correctness of artist labels. A large number of the songs on IA were restricted to 30-second duration due to copyright concerns. We preferred the full duration versions of a song, whenever available. Using these criteria, we downloaded a total of $10,063$ SOI for $1,341$ AOI.

• **SOI pruning for removing non-singing audios:** Voice Activation Detection (VAD) [9] was used on the SOI to remove silent segments. The VAD processed songs were then manually verified to discard audio files that did not contain singing vocals. Note that the human listeners only listened

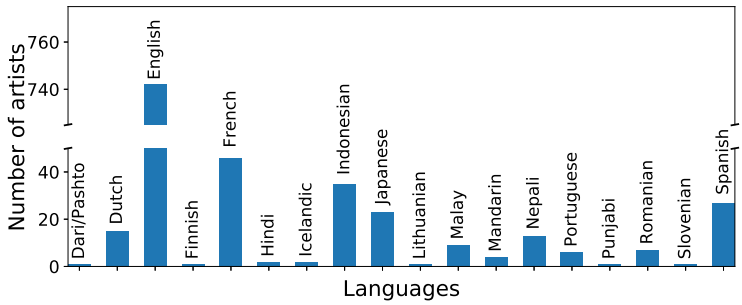


Figure 6.1: Distribution of languages in the *JukeBox* dataset

to 5 equally separated 1-second long audio segments in every song to make their decision. This process ensured a practicable manual verification process of $1,500$ hours of audio data.

- **Manual verification of language labels in non-English songs:** Nearly one-fifth of the singers in the *JukeBox* dataset are non-English singers. The language labels originally assumed the non-English singers to sing in a non-English language. However, some of the non-English singers were multilingual, and had songs in the English language as well. Therefore, a secondary manual verification of the dataset was conducted to remove English songs for non-English singers. The resulting $7,000$ SOI from 936 AOI form the *JukeBox* dataset.

- **Splitting the dataset into the train, test, and auxiliary subsets:** Finally, the set of 936 speakers in the dataset was split into three subsets (shown in Table 6.2):

– Training set: All speakers with at least three audio samples constitute the training set (670 subjects). This set is reserved for training or fine-tuning speaker recognition models.

– Test set: All speakers with exactly two audio samples constitute the test set (98 subjects). This set is reserved for evaluating trained speaker recognition models on singing voice data.

– Auxiliary set: All speakers with only one audio sample constitute the auxiliary set (168 subjects). This set can be used to augment the training data for speaker recognition models trained in the identification mode. However, the auxiliary set cannot be used to train models in the verification mode, as at least 2 samples per subject are needed to form a genuine pair.

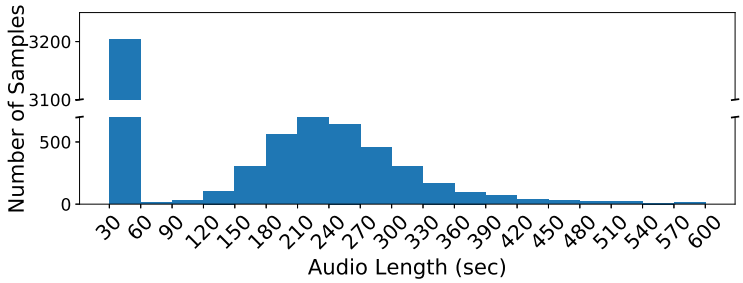


Figure 6.2: Distribution of audio length in the *JukeBox* dataset

Table 6.2: Dataset statistics of the *JukeBox* dataset

Dataset	**Train**	**Test**	**Auxillary**
# of Subjects	670	98	168
# of Male Subjects	397	57	79
# of Non-English Subjects	104	21	69
# of Samples	6,636	196	168
# of Hours	385	33	49
Max # of Samples/Speaker	87	2	1
Min # of Samples/Speaker	3	2	1
Avg # of Samples/Speaker	10	2	1

6.3 Datasets and Experimental Protocols

We propose several experimental protocols for establishing baseline speaker recognition performance on the *JukeBox* dataset. We use state-of-the-art and baseline speaker recognition methods, viz., 1D-Triplet-CNN [42], xVector-PLDA [154], and iVector-PLDA [63] for this purpose. We also evaluate their performance on the *JukeBox* dataset under different conditions based on gender of the artists and language of the songs.

6.3.1 Datasets

6.3.1.1 VoxCeleb2 Dataset

We use the VoxCeleb2 [43] dataset to perform baseline speaker recognition experiments on spoken voice data (i.e. spoken-to-spoken scenario). We use a subset of the VoxCeleb2 dataset to keep the experiments computationally tractable. A random subset of $5,994$ video samples corresponding to the $5,994$ celebrities in the VoxCeleb2 dataset forms the training set. Similarly, a random subset of 118 video samples corresponding to 118 celebrities forms the evaluation set. Speech from

each video in the dataset is extracted and split into multiple non-overlapping 5-second long audio samples.

6.3.1.2 *JukeBox* Dataset

Data from *JukeBox* dataset is used to *fine-tune* and evaluate the aforementioned speaker recognition methods on singing voice data (i.e. both spoken-to-singing and singing-to-singing scenarios). Each song in the training set was split into multiple non-overlapping 30-second long segments to increase the number of training samples. In all our experiments, we use the samples from the training set to train the speaker verification algorithms, and the samples from the test set to evaluate the performance of the trained speaker verification models.

6.3.2 Experimental Protocol

6.3.2.1 iVector-PLDA based speaker verification experiments

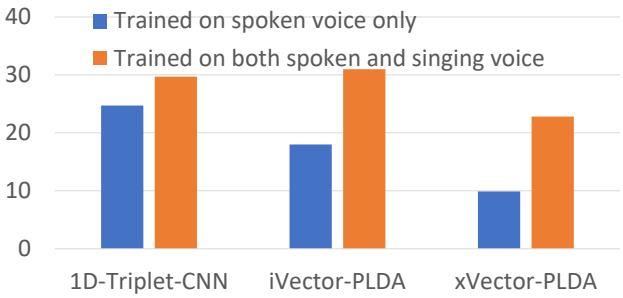
We use the MSR Identity Toolkit's [143] implementation of the iVector-PLDA algorithm as our first baseline speaker verification method. A Gaussian-PLDA (gPLDA)-based matcher [143] is used to compare the extracted i-Vector embeddings of a pair of speech samples.

6.3.2.2 xVector-PLDA based speaker verification experiments

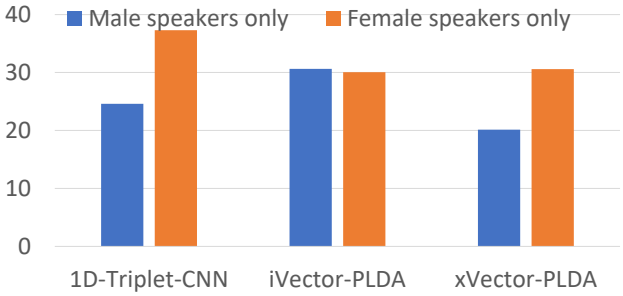
We use the PyTorch-based implementation [42] of the xVector algorithm as our second baseline speaker verification method. A gPLDA-based matcher [143] is used to compare the extracted xVector embeddings of a pair of speech samples.

6.3.2.3 1D-Triplet-CNN based speaker verification experiments

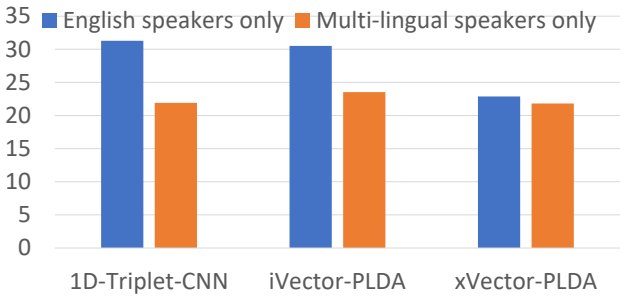
We also perform speaker verification experiments using the 1D-Triplet-CNN algorithm, due to its demonstrated robustness to audio degradations [42]. The audio samples in the training set are grouped into triplets to train the 1D-Triplet-CNN algorithm. For evaluation, the audio samples



(a) Effect of training data on singer verification.



(b) Effect of gender on singer verification.



(c) Effect of language on singer verification.

Figure 6.3: Summary of verification performance (TMR@FMR=1%) across different evaluation conditions on the *JukeBox* dataset.

are grouped into pairs and processed by the trained model to generate pairs of 1D-Triplet-CNN embeddings. These pairs of embeddings are then matched using the cosine similarity metric.

6.3.2.4 Studying the effect of gender on speaker verification

The fundamental physiological differences between male and female voices [104] have been used to advocate for their separate treatment in the context of speaker recognition [96]. These differences are further pronounced in the singing voice [160]. Male singers, for example, exhibit a larger variation in their falsetto (a method of voice production) [169], potentially making them harder to

Table 6.3: Speaker verification results on spoken voice data from the VoxCeleb2 dataset using the 1D-Triplet-CNN **[M1]**, iVector-PLDA **[M2]**, and xVector-PLDA **[M3]** models. The same models are evaluated on the *JukeBox* dataset to compare the performance on singing voice data. Here, **P1** = VoxCeleb2 , **P2** = *JukeBox* , and **P3** = Both VoxCeleb2 and *JukeBox* together.

Exp. #	Train Set /Test Set	Models	TMR @FMR=1%	minDCF	EER (in %)
1	P1/P1	M1	91.23	1.82	4.09
2		M2	92.79	1.38	3.81
3		M3	65.06	4.15	7.89
4	P1/P2	M1	**24.72**	**8.35**	26.48
5		M2	18	8.99	**24.49**
6		M3	9.9	9.56	31.83
7	P3/P2	M1	29.71	7.91	24.36
8		M2	**30.98**	**7.77**	**23.63**
9		M3	22.82	8.42	26.39

Table 6.4: Verification results on the gender and language specific evaluation subsets of the *JukeBox* dataset using the 1D-Triplet-CNN **[M1]**, iVector-PLDA **[M2]**, and xVector-PLDA **[M3]** methods. All the models were trained on the VoxCeleb2 dataset and fine-tuned using the *JukeBox* dataset. Here, **C1** = male speakers only, **C2** = female speakers only, **C3** = English speakers only, and **C4** = non-English speakers only.

Exp. #	Models	Evaluation Condition	TMR @FMR=1%	minDCF	EER (in %)
10	M1	C1	24.6	8.33	24.44
11		C2	37.29	6.4	21.95
12		C3	31.28	7.67	21.7
13		C4	21.91	8.18	33.63
14	M2	C1	30.64	7.87	26.41
15		C2	30.05	7.58	22.43
16		C3	30.51	7.75	23.67
17		C4	23.53	7.67	28.48
18	M3	C1	20.14	8.57	25.09
19		C2	30.59	7.72	29.29
20		C3	22.88	8.41	24.72
21		C4	21.81	8.44	38.96

recognize than their female counterparts. Therefore, in this work, we perform gender-specific speaker verification experiments (Exp. # 10, 11, 14, 15, 18, and 19 in Table 6.4) to study the effect of gender on speaker verification from singing voice data. We use the following two types of gender-specific trials in our experiments:

Female only trials: In these experiments, the trained models are evaluated on same-gender (fe-

male only) trials drawn from 41 female artists in the test set of the *JukeBox* dataset.

Male only trials: In these experiments, the trained models are evaluated on same-gender (male only) trials drawn from 57 male artists in the test set of the *JukeBox* dataset.

6.3.2.5 Studying the effect of language on speaker verification

Speaker recognition performance of both humans and machines degrade when the speech audio being evaluated is in a language unknown or unfamiliar to the listener [99]. This is also known as the language-familiarity effect (LFE) [61]. In this work, we perform additional speaker verification experiments on the *JukeBox* dataset to evaluate the effect of language on speaker verification performance from singing audio. We perform two different types of language-based speaker verification experiments, given by Exp. # 12, 13, 16, 17, 20, and 21 in Table 6.4 and described below. All the models in this set of experiments were trained and fine-tuned using the multilingual speech data from the VoxCeleb2 and the *JukeBox* datasets, respectively.

Same language, English only trials: In these experiments, the models are evaluated on same-language (English only) trials drawn from 77 English singers in the test set of *JukeBox*.

Multilingual, non-English trials: In these experiments, the models are evaluated on multilingual trials drawn from 21 non-English singers in the test set of *JukeBox*. The songs in the multilingual trials are sung in one of these 9 different non-English languages: Dari/Pashto, Dutch, French, Japanese, Mandarin, Nepali, Punjabi, Romanian, Spanish.

6.3.2.6 Studying the effect of singing style modeling on speaker verification

Finally, we also perform a fusion of Global Style Token (GST) [168] based prosodic speech features with the 1D-Triplet-CNN based speaker embedding to facilitate singing style modeling for speaker verification. In these experiments, we extract the speaker embeddings obtained from the 1D-Triplet-CNN and input it to GST to extract prosodic speech features. These prosodic speech

Table 6.5: Effect of prosody modeling for singing-style based speaker recognition. The 1D-Triplet-CNN + GST model performs singing-style based speaker recognition. The numbers represent performance when trained on the VoxCeleb2 dataset only / on both the VoxCeleb2 and the *JukeBox* datasets

Models	TMR@FMR=1%	minDCF	EER (in %)
1D-Triplet-CNN	**24.72/29.71**	**8.35/7.91**	**26.48**/24.36
1D-Triplet-CNN + GST	19.42/26.80	8.78/8.24	26.55/**24.27**

features are further fused with the 1D-Triplet-CNN based speaker embeddings to derive a style-sensitive speaker embedding. This embedding is then used to perform speaker verification experiments, given in Table 6.5.

6.4 Results and Analysis

The results of all the experiments described in Section 7.4.2 are given in Tables 6.3, 6.4, and 6.5, and Figure 6.3. For all the speaker verification experiments, we report the True Match Rate at a False Match Rate of 1% (TMR@FMR=1%), minimum Detection Cost Function (minDCF) and Equal Error Rate (EER in $\%$). The minimum Detection Cost Function (minDCF) is computed at a prior probability of 0.01 for the specified target speaker (P_{tar}) with a cost of missed detection of 10 (C_{miss}).

- In the experiments 1 to 3 given in Table 6.3, baseline speaker verification performance is established for all the models on spoken voice data from the VoxCeleb2 dataset. The relatively lower performance of the xVector-PLDA model is attributed to the limited training data being insufficient for learning xVector-PLDA model's considerably larger parameter space.

- Further, in experiments 1 to 6, a large performance drop is noted across all models when they are evaluated on the *JukeBox* dataset when compared to the VoxCeleb2 dataset. This indicates the difficulty of performing singer recognition using models that are pre-trained on spoken voices.

- Fine-tuning the models pre-trained on the VoxCeleb2 dataset, using the training set of *JukeBox* (in experiments 7 to 9) improved the average performance (TMR@FMR=1%) of all the models by $\sim 10.29\%$. This indicates the benefit of using *JukeBox* for fine-tuning pre-trained speaker

recognition models for the task of singer recognition.

- We also performed speaker identification experiments corresponding to the experimental protocol given in Table 6.3. The identification results follow the trend seen in verification. Best performance is observed when the models are trained and tested on spoken voice. Worst performance is observed when the models are trained on spoken voice and tested on singing voice. Fine-tuning the models trained on spoken voice with singing voice improves the performance on singing voice.

- In the gender-based speaker verification experiments (10, 11, 14, 15, 18, and 19) given in Table 6.4, majority of the models perform better on female subjects. This is an interesting result because (a) both the VoxCeleb2 and *JukeBox* datasets have a higher proportion of male subjects in the training data, and (b) gender-based speaker recognition experiments on spoken speech data usually perform better for males [96, 104]. This demonstrates the effect of gender-specific voice range profiles of the singing voice [160] in the context of speaker recognition.

- In the language-based speaker verification experiments (12, 13, 16, 17, 20, and 21) given in Table 6.4, majority of the models perform better on English-only trials. This indicates the presence of the LFE even in singing audios, where the speaker models trained on English-majority speech data performs better on English-only speech data compared to non-English speech.

- The inclusion of prosody modeling for encoding the singing style in the speaker embeddings degrades the speaker verification performance (see Table 6.5). This can be attributed to the large intra-speaker variance due to different singing styles used in different songs. This indicates that the singing-style of the singer estimated from a fixed set of songs does not generalize well across other songs, leading to a drop in performance.

6.5 Summary

We assembled a multilingual singer recognition dataset called *JukeBox*. The evaluation of state-of-the-art speaker recognition methods trained only on spoken voice data, on the *JukeBox* dataset, revealed the challenges posed by singing voice data to speaker recognition. The *JukeBox* dataset

can be used to address these challenges by facilitating speaker recognition research on singing voice data. Additionally, the dataset is annotated for language and gender labels, which can be used to investigate their effects on singer recognition performance. In the future, we plan to extend this dataset to include spoken voice audios for the singers in the current dataset. This will help us study the relationship between the spoken voice and the singing voice of a subject, in the context of speaker recognition.

CHAPTER 7

SINGING VERSUS SPOKEN VOICE: DOMAIN ADAPTATION FOR SPEAKER RECOGNITION

Portions of this chapter appeared in the following publication:

Chowdhury, Anurag, Cozzo, Austin, and Arun Ross. "Singing Versus Spoken Voice: Multi-task Domain Adaptation for Speaker Recognition" INTERSPEECH (2021 - Submitted).

7.1 Introduction

In the previous chapter, we assembled a multi-lingual singing voice dataset, called JukeBox, for facilitating the development and evaluation of speaker recognition methods on the unique challenges of the singing voice. Specifically, we studied the problem of speaking style variability across spoken and singing voices and its effect on speaker recognition performance. Additionally, we also noted the challenges of performing speaker recognition in the singing voice include increased intra-user variance due to increased vocal range of the singing voice and a wide variety of background noise such as background chorus and instrumentation. This chapter continues studying the effect of speaking style and audio condition variability between the spoken and singing voice on speaker recognition performance. Specifically, we propose using domain adaptation to develop speaker recognition methods robust to varying speaking styles and audio conditions. Domain adaptation is observed to improve the speaker recognition performance (true match rate at a false match rate of 1%) by over 12% and 2% for the singing and spoken voice, respectively. A detailed analysis of the domain-adapted method's speech embeddings explains its generalizability across varying speaking styles and audio conditions. Finally, we also extend the singing voice data in the JukeBox dataset with corresponding speaking voice data and refer to it as JukeBox-V2. This extended dataset is assembled for facilitating evaluation and future development of cross-modal speaker verification methods (i.e., compare singing voice to the spoken voice of a speaker).

Speaker recognition, or voice biometrics, entails comparing two speech samples to determine if the same individual produced them. Most speaker recognition systems assume 'ideal audio conditions,' such as minimal background noise, neutral speaking style, and normal vocal effort for optimal performance [91]. However, such an assumption is an oversimplification of practical voice biometrics scenarios. While several recently developed methods have focused on performing speaker recognition in the presence of background noise and degradations, the majority of them only consider spoken voice (i.e., speech uttered in a neutral speaking style) for training and evaluating their approaches [42, 91]. Spoken voice, however, only represents a limited range of possible vocal dynamics for a speaker [70]. Therefore, methods based on neutral spoken voice suffer performance degradation with varying speaker style and effort [152].

Among possible speaking styles, singing voice presents a particularly less explored mode of speaker recognition [111]. The challenges of singer recognition – speaker recognition where the speaking style is singing – differs from traditional speaker recognition due to the much broader range of perceptual qualities and underlying physiological dynamics apparent in the singing voice [30, 31, 45]. The singing voice's features are further diversified by the singing style, which is influenced by the genre and accompanying music [163]. Singing voice, thus, serves as a surrogate for a wide variety of speaking styles and audio conditions that present a challenge to traditional speaker recognition systems [112].

A recent work assembled a singing voice dataset, JukeBox [39], and demonstrated the challenges of performing singer recognition using models pre-trained on spoken voice. Furthermore, the pre-trained models were fine-tuned using singing voice to improve singer verification performance. However, the fine-tuned models were not evaluated on spoken voice to determine their generalizability across different speaking styles. In addition, the original JukeBox dataset does not contain any spoken voice samples corresponding to a person's singing voice, limiting its utility for cross-domain speaker verification, i.e., matching a person's singing voice to their spoken voice.

Following these observations, the contributions of this work are as follows. We first extend the original JukeBox dataset to include spoken voice samples for the subjects in the evaluation set

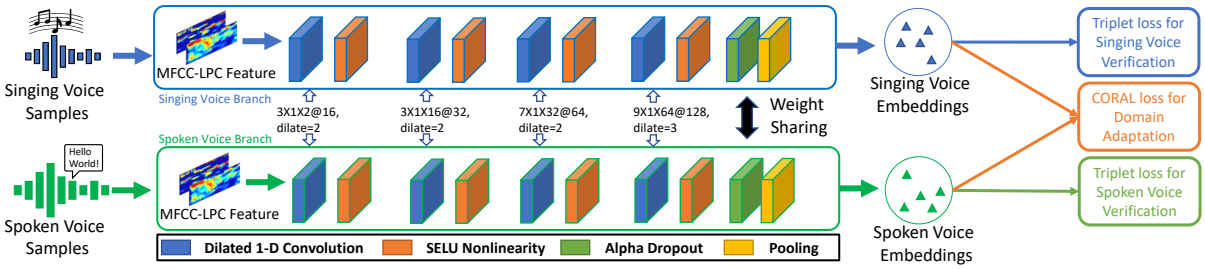


Figure 7.1: A visual representation of the domain-adaptation-based 1D-CNN framework proposed in Section 7.3.

of the JukeBox dataset. This extended dataset, referred to as JukeBox-V2, enables the evaluation of speaker recognition methods across varying speaking styles and which we make publically available.[1] We next incorporate domain adaptation in speaker recognition system to learn a highly-discriminative feature space that equitably represents both singing and spoken voice data, thereby providing generalizable performance across the two speaking styles. Finally, we also analyze the impact of variation in speaking style (in this case, singing versus spoken voice), on the learnt feature space.

7.2 Motivation

The original study on the JukeBox dataset *fine-tuned* the pre-trained speaker recognition models using singing voice for improving singer recognition performance [39]. However, as we will demonstrate in this paper, these fine-tuned models result in performance degradation on spoken voice (as noted in Section 7.5 and Fig. 7.2). Therefore, the performance degradation on *spoken* voice accompanied by a modest increase in performance on *singing* voice suggests that fine-tuning is a suboptimal solution and that the problem has to be carefully revisited.

This disparity in acoustical characteristics of the singing voice with respect to the spoken voice is similar to another form of speech - the whispering voice. For example, the F1 and F2 formants of the whispering [78] and singing voices [70] deviate from the spoken voice. Due to such intrinsic variations in the acoustic characteristics, the whispering voice is often treated as a speaking style variation characterized by a low vocal effort and unvoiced speech [57, 78]. Similarly, as done

[1] http://iprobe.cse.msu.edu/datasets/jukebox_v2.html

in [39, 112] and this work, singing voice is assumed to be a speaking style variation characterized by an increased vocal range.

Speaker recognition systems are often adversely affected by a wide variety of perturbations, both extrinsic and intrinsic. While extrinsic perturbations such as channel variability are often addressed by techniques such as dataset variability compensation [**?**], intrinsic perturbations such as language and vocal effort variability are often resolved using domain adaptation (DA) [**?**, **?**]. Therefore, in this work we use unsupervised DA to develop speaker recognition methods robust to speaking style variability. Specifically, we use the CORAL [161] and DeepCORAL [162] techniques due to their simplicity and demonstrated effectiveness for bridging the domain gap created by intrinsic variabilities in the human voice [**?**]. To the best of our knowledge, this is the first work to explore DA for developing speaker recognition models robust to speaking style variabilities.

7.3 Domain Adaptation-based Speaker Recognition Framework

Speaker-dependent speech features such as phoneme duration, mean fundamental frequency (F0), and formant center frequencies that are crucial for speaker modeling differ vastly between the speaking and the singing voice, thus creating a **domain gap** between the two speaking styles [70, 78]. Therefore, in this work, we develop a DA-based speech encoding framework for reducing the domain gap between the singing and speaking voice, so as to improve speaker recognition performance on singing voice while minimizing the loss of performance on speaking voice. Toward that end, we design the DA-based 1D-CNN framework as shown in Fig. 7.1. The proposed framework uses a 1D-Triplet-CNN [42] to extract speech embeddings from both speaking and singing voice. The distance between the covariances of the speaking and singing voice embeddings is then minimized using the CORAL loss [161, 162] in order to bridge the domain gap between the two speaking styles. Similarly, we also use CORAL loss to domain-adapt the probabilistic linear discriminant analysis (PLDA) classifiers in the iVector-PLDA [50] and xVector-PLDA [154] methods. This will help the effectiveness of domain adaptation in both classical and state-of-the-art speaker recognition methods.

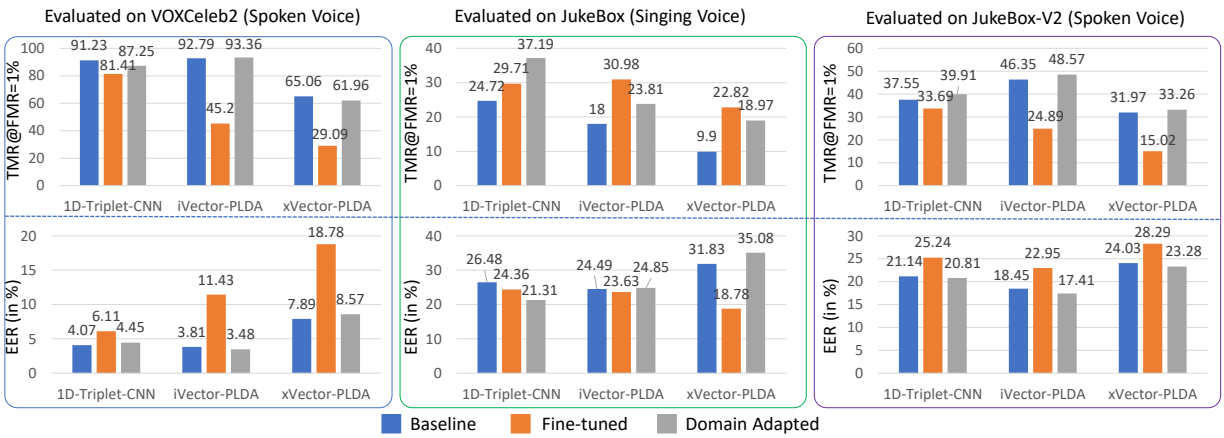


Figure 7.2: Summary of verification performance (Top: TMR@FMR=1%, Bottom: EER (in%)) across different evaluation conditions. Note the increase in singer recognition performance in both fine-tuned (orange bars) and domain adapted (grey bars) models and increase in speaker recognition performance in domain adaptation over fine-tuning.

The proposed framework (Fig. 7.1) consists of two identical 1D-CNN [42] branches with shared weights for processing singing (x_{si}) and spoken (x_{sp}) voice samples separately. Each branch extracts an MFCC-LPC feature patch [42] from the input audio and feeds it to the 1D-CNN to extract corresponding speech embeddings, $g(x_{si})$ and $g(x_{sp})$. The speech embeddings along with their speaker labels are passed to an adaptive triplet mining technique [41] to process corresponding speech triplets for singing $(S_a^{si}, S_p^{si}, S_n^{si})$ and spoken $(S_a^{sp}, S_p^{sp}, S_n^{sp})$ speech samples. Here, S_a^{sp} and S_p^{sp} are the *anchor* and the *positive* spoken voice samples from a subject X. S_n^{sp} is the *negative* spoken voice sample from another subject Y. The two set of triplets from the singing and spoken voice data are then used to minimize the corresponding cosine triplet embedding losses [42], L_{si} and L_{sp}, for training the 1D-CNN branches. The functional form of both the losses is given by:

$$L(S_a, S_p, S_n) = \sum_{a,p,n}^{N} \cos(g(S_a), g(S_n)) - \cos(g(S_a), g(S_p)) + \alpha_{margin} \tag{7.3.1}$$

Here, N is the total number of triplets drawn by the adaptive triplet mining method [41]. α_{margin} is the margin of the minimum distance between positive and negative samples and is a user-tunable hyper-parameter.

For performing DA between the singing and spoken voice samples, we minimize the distance between the covariances C_{si} and C_{sp}, known as the CORAL loss [161, 162], of the singing and spoken voice embeddings $g(x_{si})$ and $g(x_{sp})$. The DA loss (L_{DA}) is given by:

$$L_{DA}(g(x_{si}), g(x_{sp})) = \frac{1}{4d^2} \left\| (C_{si} - C_{sp}) \right\|_F^2 \tag{7.3.2}$$

Here, $\|\cdot\|_F^2$ is the squared matrix Frobenius norm and d is the dimensionality of the voice embeddings. The combined loss L for the entire framework is given as follows:

$$L = \alpha_1 L_{si} + \alpha_2 L_{sp} + \beta L_{DA} \tag{7.3.3}$$

Here, α_1, α_2, and β are user-tunable hyper-parameters (in our experiments, 1, 10, and 10, respectively) that control the effect of individual losses on the combined loss. For evaluation, speech embeddings extracted using the trained model are matched using the cosine similarity metric.

7.4 Datasets and Experimental Protocols

7.4.1 Datasets

• VoxCeleb2 Dataset: We use the VoxCeleb2 [43] dataset to train and evaluate the speaker recognition models on spoken voice data (i.e., spoken-to-spoken scenarios). Similar to [39], we use a subset of $5,994$ video samples corresponding to the $5,994$ celebrities in the VoxCeleb2 dataset to form the training set. One video per subject is selected to ensure that the effects of fine-tuning are noteworthy. While more speaking voice data would improve performance on speaking voice evaluations (particularly in the case of xVector-PLDA), similarly sized source (spoken voice) and target (singing voice) domain datasets are important for effective DA. A random subset of 118 video samples corresponding to 118 celebrities forms the evaluation set. Speech from each video sample is split into multiple non-overlapping 5-second long audios.

• JukeBox Dataset: We use the JukeBox [39] dataset to train/fine-tune and evaluate the speaker recognition models on singing voice data (i.e., both spoken-to-singing and singing-to-singing scenarios). Training data is augmented by splitting each song into multiple non-overlapping 30-second long segments. Furthermore, following the data collection protocol in [39], we collected four 5-second long spoken voice samples corresponding to 92 out of 98 subjects in the evaluation set of the JukeBox dataset. We could not locate any spoken voice data for the remaining six subjects. We collected the data by identifying interviews of each singer on YouTube and manually isolating the

target's speech audio. This extension of the JukeBox dataset, referred to as JukeBox-V2, enables cross-domain evaluation of speaker recognition algorithms.

7.4.2 Experiments Performed

We perform multiple experiments (Fig. 7.2 and Table 7.1), listed below, to evaluate the effect of domain adaptation on the iVector-PLDA [50], xVector-PLDA [154], and 1D-Triplet-CNN [42] algorithms for improving their robustness to speaking style variabilities. We follow the experimental protocols given in [39] to compare the performance of domain adapted models with (a) baseline models (trained only on spoken voice data), and (b) fine-tuned models (trained on spoken voice data and fine-tuned on singing voice data).

- Spoken voice recognition experiments: Here, speaker verification performance is evaluated on spoken voice from the VoxCeleb2 and JukeBox-V2 datasets (shown in Fig. 7.2).
- Singing voice recognition experiments: Here, speaker verification performance is evaluated on singing voice from the JukeBox dataset (shown in Fig. 7.2).
- Cross-domain voice recognition experiments: Here, cross-domain speaker verification performance is evaluated by matching singing voice to spoken voice from the JukeBox and JukeBox-V2 datasets, respectively (shown in Table 7.1).

7.5 Results

We report the performance using two metrics: True Match Rate at a False Match Rate of 1% (TMR@FMR=1%) and Equal Error Rate (EER in $\%$).

- All the baseline models attain significantly lower performance on singing voice than their spoken voice counterpart in the JukeBox-V2 dataset. This reinforces the challenges faced by models trained on spoken voice when evaluated on the singing voice [39].
- As also noted in [39], the xVector-PLDA model's relatively lower performance is attributed to the training data being insufficient for training xVector's considerably larger parameter space

Table 7.1: Poor speaker verification results on cross-modal voice data from the JukeBox-V2 dataset justifying the application of DA (see Fig. 7.2)

Models	Variant	TMR@FMR=1%	EER (in %)
1D-Triplet-CNN	Baseline	1.39	48.17
	Finetuned	0.6	43.02
	DomainAdapted	1.67	42.11
iVector-PLDA	Baseline	2.37	49.11
	Finetuned	1.36	43.54
	DomainAdapted	1.73	44.64
xVector-PLDA	Baseline	0	48.14
	Finetuned	1.2	44.82
	DomainAdapted	1.58	42.78

(4.2M) compared to the 89K parameters in the 1D-Triplet-CNN and an even lower parameter space in the iVector-PLDA model.

- On average, the fine-tuned models, compared to the baseline models, demonstrate an increase in performance (TMR@FMR=1%) on singing voice by $\sim 10\%$, but they also demonstrate an average performance loss on spoken voice in the JukeBox-V2 and the VoxCeleb2 datasets by $\sim 14\%$ and $\sim 31\%$, respectively. This demonstrates the fine-tuned models' lack of generalizability across speaking styles.

- The DA method outperforms (TMR@FMR=1%) the corresponding baseline models on singing voice from the JukeBox-V2 dataset by $\sim 9\%$ on average. It also reduces the average performance loss on spoken voice in the VoxCeleb2 dataset from $\sim 31\%$ to $\sim 2\%$. Furthermore, in the spoken voice JukeBox-v2 dataset, the average performance loss of $\sim 14\%$ is converted to a performance *gain* of $\sim 2\%$. **This demonstrates domain-adapted models' generalizability across speaking styles.**

- In the cross-domain speaker recognition experiments in Table 7.1, the DA models does not offer any significant performance improvement over the baseline methods, as it is unable to map a person's spoken voice to their singing voice. Towards this end, we believe cross-domain training data (currently unavailable) is essential to learn the mapping between an individual's singing and spoken voice.

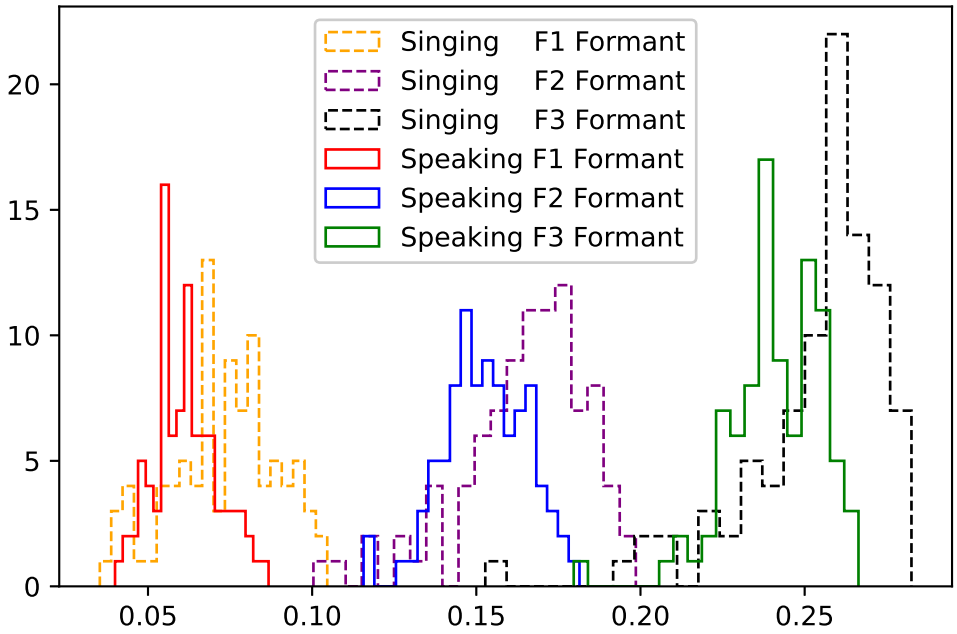


Figure 7.3: Histogram plots of the first three formants (F1-F3) of spoken and singing speech from the JukeBox-V2 dataset

7.6 Analysis

In Section 7.5, we experimentally demonstrated the domain gap between the singing and spoken voice. A histogram of the first three formants of the singing and spoken voice in the JukeBox-V2 dataset qualitatively verified the presence of the domain-gap in Figure 7.3. In this section, we inspect the effect of this domain gap in the feature space learned by a 1D-Triplet-CNN-, iVector-PLDA-, and xVector-PLDA-based baseline speaker recognition models, trained on spoken voice alone. In comparison, we analyze the feature space learned by the different approaches when combined with DA to understand the effect of DA across the two speaking styles. To this end, we compare the t-SNE [100] plots of speech embeddings of spoken and singing voice data from the JukeBox-V2's evaluation set, extracted by the 1D-Triplet-CNN, iVector-PLDA, and xVector-PLDA-based models, both without and with DA (Fig. 7.4). In Fig. 7.4, the singing and spoken voice embeddings extracted by the models without DA form separate clusters demonstrating the presence of the domain gap. **However, in the models with DA, the clustering is reduced or even eliminated.** This difference in the speech embedding clusters between the DA and non-DA mod-

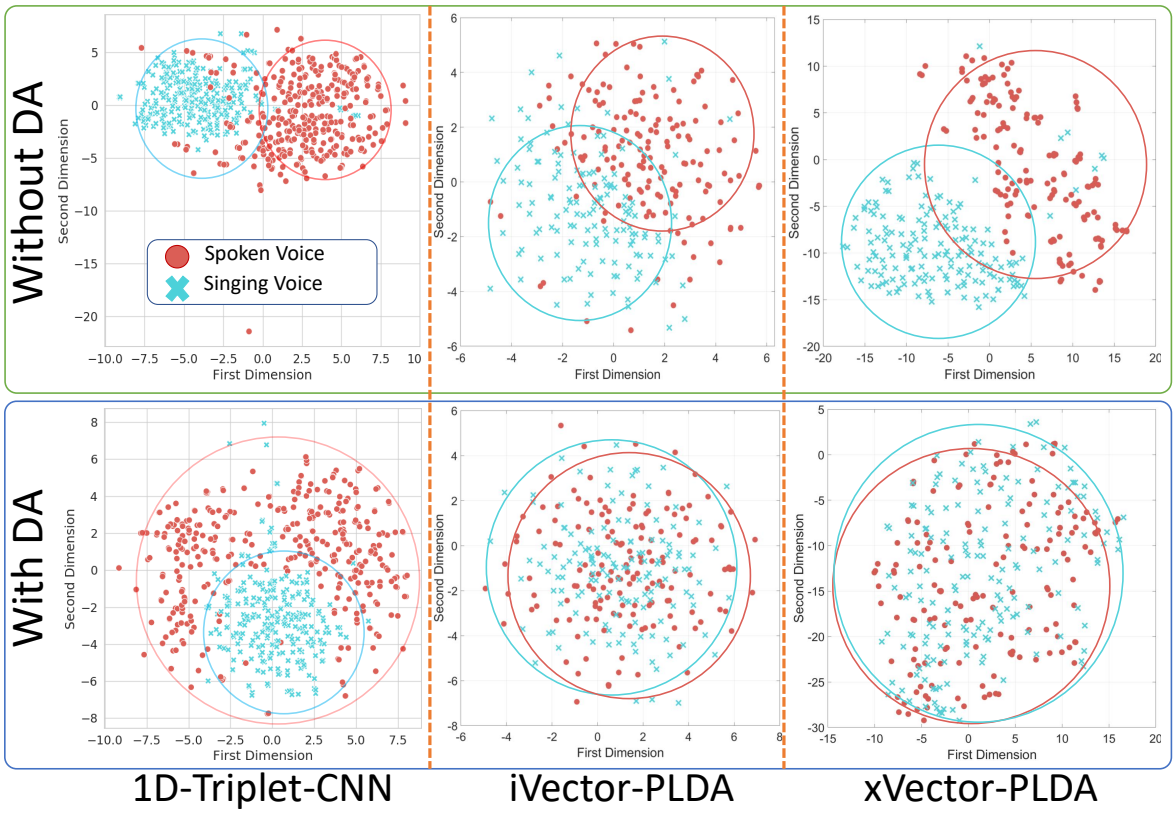


Figure 7.4: t-SNE plots of the speech embeddings (with and without DA) of singing and spoken voice from the JukeBox-V2 dataset. The circles were added to indicate the apparent cluster boundaries. After DA, the domain gap is reduced leading to overlap of the circle as shown in the lower row.

els is an effect of the CORAL and DeepCORAL loss used for performing DA. The DA minimizes the covariance between the speech embeddings of the two speaking styles, thereby merging their clusters and bridging the domain gap.

7.7 Summary

Singing voice data introduces the challenges of varying speaking style and background noise to speaker recognition. Therefore, training speaker recognition models using domain adaptation, as evidenced by the experiments, has the potential to improve their generalizability across varying speaking styles. We also assembled the JukeBox-V2 dataset to demonstrate the challenges of cross-domain speaker recognition. Toward that end, it may be valuable to explore the benefits of combining speaking style-specific speech filter banks [173] with domain-adaptation to

improve cross-domain speaker recognition performance. Additionally, the availability of cross-domain training data is important to develop cross-domain speaker recognition systems.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Research Contributions

Speaker recognition, also known as voice or talker recognition, is recognizing an individual from their voice. Historically, human psychology and medicine were the first disciplines to study the various mechanisms governing the production and perception of the human voice in the context of speaker recognition [54, 55]. They explored the various factors of variability in the human voice, such as pitch, intensity, time, timbre, and volume. As a result, the improved understanding of the human voice's uniqueness motivated further research to assess its reliability as a biometric modality in a wide variety of applications [108], including as legal evidence [36]. Later in the 20th century, the advent of digital signal processing techniques powered by digital computers led to automated machine-driven techniques for performing speaker recognition [71]. Some of the early research works studied the effects of variations in the time-frequency-energy features on speaker recognition performance [23, 75, 131]. This showcased the importance of identifying speech features best suited for performing speaker recognition. While the techniques used for extracting speaker-dependent speech features have significantly changed over the years, the focus of speaker recognition research has still largely remained on the discovery of new and robust speaker-dependent speech characteristics. Specifically, there has been an increased research effort to discover speech features and corresponding feature extraction techniques that are robust to the covariates of non-ideal audio conditions, such as background noise, short audio duration, language, and speaking style variability.

In this thesis, several deep learning-based techniques were developed for performing robust speaker recognition from audio samples collected in diverse acoustic environments and exhibiting a wide variety of intrinsic speech variabilities. Specifically, deep learning-based techniques were developed for extracting speaker embeddings robust to extrinsic factors such as background

noise, reverberation, varying-duration speech, and intrinsic factors, such as variability of language and speaking style. First, a 1-dimensional convolutional neural network (1D-CNN) that uses 1D filters, rather than 2D filters, was developed for extracting noise-robust speech embedding from cepstral speech features, such as the Mel-frequency Cepstral Coefficients (MFCC). The 1D-CNN filters were designed to learn inter-dependency between cepstral coefficients extracted from audio frames of fixed temporal expanse. Also, the 1D-CNN was designed to extract speech embeddings independently from each input audio frame and retain only the embeddings that were common across several input audio frames. This approach was essential for reliably extracting noise-robust speech embeddings due to its focus on extracting speaker-dependent speech features that were consistent across multiple frames.

Further, the 1D-CNN architecture was extended to judiciously combine two commonly used speech features: Mel Frequency Cepstral Coefficients (MFCC) and Linear Predictive Coding (LPC) at feature-level. Such a combination of MFCC and LPC features allowed the network to encode speech perception and speech production characteristics into highly-discriminative and robust speaker embeddings. This extended 1D-CNN architecture was further trained using a triplet learning framework to combine these two features in a novel manner, enhancing speaker recognition performance in challenging scenarios. Therefore, the extended 1D-CNN architecture was named 1D Triplet Convolutional Neural Network (1D-Triplet-CNN). Extensive evaluation on multiple datasets, different types of audio degradations, multi-lingual speech, and varying length of audio samples conveyed 1D-Triplet-CNN's efficacy over existing state-of-the-art speaker recognition methods, such as iVector-PLDa and xVector-PLDA, in severely degraded speech samples.

The majority of automatic speaker recognition algorithms, including the proposed 1D-CNN and 1D-Triplet-CNN methods, use pre-defined filterbanks, such as Mel-Frequency and Gammatone filterbanks, for characterizing speech audio. The design of these filterbanks is based on domain-knowledge and limited empirical observations [153, 176]. The resultant features often do not generalize well to a wide variety of audio degradations. Therefore, a deep learning-based technique was proposed to induce the design of a filterbank from vast amounts of speech audio.

The purpose of such a filterbank is to extract features that are robust to degradations in the input audio. To this effect, a 1D convolutional neural network (1D-CNN) is designed to learn a time-domain filterbank, called DeepVOX, directly from raw speech audio. An adaptive triplet mining technique was also developed to efficiently mine the data samples best suited to train the filterbank. A detailed ablation study of the DeepVOX filterbanks revealed the presence of both vocal source and vocal tract characteristics in the extracted features. This could be seen as a direct effect of the presence of both magnitude and phase information in the raw input speech audio, as magnitude information in speech audio captures vocal tract features and phase information captures rich vocal source information [86]. Experimental results on VOXCeleb2, NIST SRE 2008 and 2010, and Fisher speech datasets demonstrate the efficacy of the DeepVOX features across various audio degradations, multi-lingual speech data, and varying-duration speech audio. The DeepVOX features also improved existing speaker recognition algorithms' performance, such as the xVector-PLDA and the iVector-PLDA. Therefore, DeepVOX can be used to directly replace handcrafted features in currently deployed speaker recognition methods and potentially improve their performance.

Automatic speaker recognition algorithms typically characterize speech audio using short-term spectral features, such as MFCC and LPC, that encode the physiological and anatomical aspects of speech production. However, such algorithms do not fully capitalize on speaker-dependent characteristics present in behavioral speech features. Therefore, a prosody encoding network called DeepTalk was proposed for extracting vocal style features directly from raw audio data. The DeepTalk method outperformed several state-of-the-art speaker recognition systems across multiple challenging datasets. Further, DeepTalk was combined with state-of-the-art physiological speech feature-based speaker recognition systems, such as iVector-PLDA, xVector-PLDA, and 1D-Triplet-CNN, at score-level to further improve speaker recognition performance in non-ideal audio conditions. This demonstrated the benefits of combining physiological and behavioral speech characteristics for improving overall speaker recognition performance. Furthermore, DeepTalk was also integrated into a state-of-the-art speech synthesizer to generate synthetic speech. The

synthesized speech audios were analyzed to understand the speech characteristics encoded by the DeepTalk method. The analysis showed that the DeepTalk reliably captured F0 contour characteristics, thus establishing its ability to model a person's vocal style characteristics. Furthermore, DeepTalk-based synthetic speech was shown to be almost indistinguishable from the real speech in the context of speaker recognition.

Vocal style characteristics were, therefore, observed to capture important speaker-dependent speech characteristics that can complement and consequentially improve the performance of physiological speech feature-based speaker recognition systems. However, most speaker recognition systems are trained and evaluated using spoken voice or everyday conversational voice data, which exhibits a limited range of possible speaker dynamics or vocal styles. This constrains the utility of the derived speaker recognition models to only spoken voice with a neutral speaking style. Therefore, it was important to assemble a speaker-annotated speech dataset encompassing a wide variety of speaking styles to facilitate speaker recognition research on diverse speaking styles. Towards that end, JukeBox - a speaker recognition dataset with multilingual singing voice audio annotated with singer identity, gender, and language labels was assembled to address this issue. The singing voice was chosen due to its coverage of a broader range of vocal and ambient factors, thus making it suitable to evaluate a speaker recognition system's robustness to diverse speaking styles. State-of-the-art speaker recognition methods were used to demonstrate the difficulty of performing speaker recognition on singing voice using models trained on spoken voice alone. The evaluation set of the JukeBox dataset was further extended with corresponding speaking voice data, referred to as JukeBox-V2. This extension allowed for cross-domain evaluation of the speaker recognition systems, i.e., match an individual's singing and spoken voice. The effect of audio condition variation, such as background chorus and instrumentation, between the speaking and singing voice on speaker recognition performance was also studied. Finally, a domain adaptation-based speaker recognition method robust to the speaking style variability between the spoken and singing voice and their corresponding audio conditions was developed. The proposed domain-adaptation-based method outperformed several baseline methods on both speaking and singing voice.

8.2 Future Work

This thesis focused on several important open challenges in speaker recognition and proposed methodologies to address them. However, a retrospective analysis of all the algorithms and methodologies proposed in this work reveals some key limitations. We use these limitations to chart a path toward possible future works that can extend this body of research.

1. The 1D-CNN and the 1D-Triplet-CNN were developed to extract speaker-dependent speech characteristics from audio frames of fixed temporal expanse. These speech characteristics extracted from the individual speech frames were then aggregated across multiple frames using temporal-average pooling to extract a corresponding single fixed-dimensional embedding. Such a pooling technique is extremely computationally efficient for extracting robust speech embedding, averaged across multiple frames. However, it assumed an equal contribution of each frame towards the aggregate representation. Such an assumption is sub-optimal in practical scenarios as their contribution to the net aggregate should be weighed depending on the amount of nuisance factors present in the corresponding frame. Towards that end, we suggest replacing the temporal-average pooling in the proposed 1D-CNN-based speaker recognition models with an attention-based pooling framework. The attention-based pooling can appropriately weigh each speech frame's contribution in the aggregated representation to yield a much more effective and robust speech representation.

2. The DeepVOX architecture was developed to extract speaker-dependent speech characteristics directly from raw audio frames. However, the current architecture is limited to using audio sampled at 8000 Hz only. The sampling rate of 8000Hz was initially chosen to adhere to the audio specifications of telephony speech audio, as used in the majority of the NIST SRE challenges [1–3]. However, some of the recently collected datasets, such as VOX-Celeb [122], that acquired speech audio from web sources, such as YouTube, have propagated the use of speech audio sampled at 16000Hz. Furthermore, in some recent speaker recognition research, significant performance improvements have been noticed when using

speech audio sampled at higher rates for developing and evaluating the models [123]. This could be attributed to the presence of more sophisticated speaker-dependent speech characteristics in higher resolution audio samples. Therefore, the potential performance benefits of using higher sampling rate audio for training the DeepVOX model suggest a redesign of the input layer of the DeepVOX model to allow training on audio sampled at higher sampling rates, as a potential future extension of this research.

3. In our work on DeepTalk, we demonstrated the benefits of combining physiological and behavioral speech characteristics for improving speaker recognition performance in non-ideal audio conditions. We combined multiple physiological feature-based speaker recognition methods, such as 1D-Triplet-CNN, iVector-PLDA, and xVector-PLDA, with the DeepTalk method at score-level to demonstrate the benefits of such a combination. In the future, we would suggest a possible extension of this work by developing an end-to-end feature-level fusion framework for combining the complementary speech characteristics present in the behavioral and physiological speech features into a robust and highly-discriminative combined speaker embedding.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] 2008 NIST speaker recognition evaluation training set part 2 ldc2011s07. https://catalog.ldc.upenn.edu/LDC2011S05. Accessed: 2018-03-06.

[2] 2010 NIST speaker recognition evaluation test set ldc2017s06. https://catalog.ldc.upenn.edu/LDC2017S06. Accessed: 2018-03-06.

[3] 2018 NIST speaker recognition evaluation test set LDC2020S04. https://catalog.ldc.upenn.edu/LDC2020S04. Accessed: 2020-12-07.

[4] Acoustic theory of speech production. http://clas.mq.edu.au/speech/acoustics/frequency/source.html. Accessed: 2017-04-29.

[5] Amazon Alexa voice recognition. https://www.theverge.com/circuitbreaker/2017/10/11/16460120/amazon-echo-multi-user-voice-new-feature. Accessed: 2017-12-29.

[6] Apple accessibility. https://www.apple.com/accessibility/iphone/. Accessed: 2020-03-28.

[7] Apple security: Touch id vs. face id. https://www.intego.com/mac-security-blog/apple-security-touch-id-vs-face-id/. Accessed: 2019-03-18.

[8] Banks turning to voice recognition. http://www.bbc.com/news/business-36939709. Accessed: 2017-04-29.

[9] Google WebRTC voice activity detection. https://webrtc.org. Accessed: 2020-03-04.

[10] HSBC voice id making telephone banking safer than ever. https://www.hsbc.co.uk/1/2/voice-id. Accessed: 2017-12-29.

[11] Internet archive. https://archive.org. Accessed: 2020-03-03.

[12] Internet archive API. https://archive.org/services/docs/api. Accessed: 2020-03-04.

[13] MATLAB voice activity detection by spectral energy. https://github.com/JarvusChen/MATLAB-Voice-Activity-Detection-by-Spectral-Energy. Accessed: 2018-03-06.

[14] Personalized hey siri. https://machinelearning.apple.com/2018/04/16/personalized-hey-siri.html. Accessed: 2019-03-16.

[15] Siri background noise. https://support.apple.com/en-us/HT204389. Accessed: 2017-12-29.

[16] Spotify API. https://developer.spotify.com/documentation/web-api. Accessed: 2020-03-04.

[17] Wellsfargo voice verification. https://www.wellsfargo.com/privacy-security/voice-verification/. Accessed: 2017-12-29.

[18] Windows hello. https://www.microsoft.com/en-us/windows/windows-hello. Accessed: 2020-03-25.

[19] Andre G Adami, Radu Mihaescu, Douglas A Reynolds, and John J Godfrey. Modeling prosodic dynamics for speaker recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 4, pages IV–788. IEEE, 2003.

[20] J Allen and D Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, 1979.

[21] H Arsikere, H Gupta, and A Alwan. Speaker recognition via fusion of subglottal features and mfccs. In *Interspeech*, pages 1106–1110, 2014.

[22] H Arsikere, S Lulich, and A Alwan. Estimating speaker height and subglottal resonances using MFCCs and GMMs. *IEEE SPL*, 21(2):159–162, 2014.

[23] BS Atal. Automatic speaker recognition based on pitch contours. *The Journal of the Acoustical Society of America*, 45(1):309–309, 1969.

[24] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 2009.

[25] Adam Berenzweig, Beth Logan, Daniel Ellis, and Brian Whitman. A large-scale evaluation of acoustic and subjective music similarity measures. *Computer Music Journal*, 2003.

[26] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. The million song dataset. *International Society for Music Information Retrieval Conference*, 2011.

[27] Paul Boersma et al. Praat, a system for doing phonetics by computer. *Glot international*, 5, 2002.

[28] Steven Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE TASSP*, 1979.

[29] M Brookes et al. Voicebox: Speech processing toolbox for MATLAB. *Software, available [Mar. 2011] from www. ee. ic. ac. uk/hp/staff/dmb/voicebox/voicebox. html*, 47, 1997.

[30] W. Brown, Howard Rothman, and Christine Sapienza. Perceptual and acoustic study of professionally trained versus untrained voices. *Journal of voice : official journal of the Voice Foundation*, 14:301–9, 10 2000.

[31] WS Brown Jr, Elizabeth Hunt, and William N Williams. Physiological differences between the trained and untrained speaking and singing voice. *Journal of Voice*, 1988.

[32] D Burton. Text-dependent speaker verification using vector quantization source coding. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(2):133–143, 1987.

[33] Joseph P Campbell. Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997.

[34] William M Campbell, Joseph P Campbell, Douglas A Reynolds, Elliot Singer, and Pedro A Torres-Carrasquillo. Support vector machines for speaker and language recognition. *Computer Speech & Language*, 20(2-3):210–229, 2006.

[35] Michael J Carey, Eluned S Parris, Harvey Lloyd-Thomas, and Stephen Bennett. Robust prosodic features for speaker identification. In *ICSLP*. IEEE, 1996.

[36] Christophe Champod and Didier Meuwly. The inference of identity in forensic speaker recognition. *Speech communication*, 31(2-3):193–203, 2000.

[37] N. H. Chhayani and H. A. Patil. Development of corpora for person recognition using humming, singing and speech. In *International Conference Oriental held jointly with Conference on Asian Spoken Language Research and Evaluation*, 2013.

[38] A Chowdhury and A Ross. Extracting sub-glottal and supra-glottal features from MFCC using convolutional neural networks for speaker identification in degraded audio signals. In *IEEE International Joint Conference on Biometrics*, 2017.

[39] Anurag Chowdhury, Austin Cozzo, and Arun Ross. Jukebox: A multilingual singer recognition dataset, 2020.

[40] Anurag Chowdhury and Arun Ross. Fusing mfcc and lpc features using 1d triplet cnn for speaker recognition in severely degraded audio signals. *IEEE Transactions on Information Forensics and Security*, 2019.

[41] Anurag Chowdhury and Arun Ross. DeepVOX: Discovering features from raw audio for speaker recognition in degraded audio signals. *arXiv preprint arXiv:2008.11668*, 2020.

[42] Anurag Chowdhury and Arun Ross. Fusing MFCC and LPC features using 1D Triplet CNN for speaker recognition in severely degraded audio signals. *Transactions on Information Forensics and Security*, 2020.

[43] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv:1806.05622*, 2018.

[44] C Cieri, D Miller, and K Walker. Fisher English training speech parts 1 and 2. *Philadelphia: Linguistic Data Consortium*, 2004.

[45] Ray Daniloff, Kathy Wolf, George Larsen, and Lee Evans. Allophonic variation in spoken and sung speech. *The Journal of the Acoustical Society of America*, 96(5):3349–3349, 1994.

[46] S Davis and P Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28, 1990.

[47] Micha¨el Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. FMA: A dataset for music analysis. In *International Society for Music Information Retrieval Conference*, 2017.

[48] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. FMA: a dataset for music analysis. In *International Society for Music Information Retrieval Conference*, 2017.

[49] N Dehak. *Discriminative and generative approaches for long-and short-term speaker characteristics modeling: application to speaker verification*. PhD thesis, École de technologie supérieure, 2009.

[50] N Dehak, P Kenny, R Dehak, P Dumouchel, and P Ouellet. Front-end factor analysis for speaker verification. *IEEE TASLP*, 19(4):788–798, 2011.

[51] Peter B Denes and Elliot Pinson. *The speech chain*. Macmillan, 1993.

[52] George Doddington. Speaker recognition based on idiolectal differences between speakers. In *Seventh European Conference on Speech Communication and Technology*, 2001.

[53] George Doddington, Walter Liggett, Alvin Martin, Mark Przybocki, and Douglas Reynolds. Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the nist 1998 speaker recognition evaluation. Technical report, National Inst of Standards and Technology Gaithersburg Md, 1998.

[54] Homer W Dudley. Production of artificial speech, May 27 1941. US Patent 2,243,526.

[55] Carl I Erickson. The basic factors in the human voice. *Psychological Monographs*, 36(2):82, 1926.

[56] Johan Stefan Erkelens. *Autoregressive modelling for speech coding: estimation, interpolation and quantisation*.

[57] Xing Fan and John Hansen. Speaker identification within whispered speech audio streams. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19:1408 – 1421, 08 2011.

[58] M Fedila, M Bengherabi, and A Amrouche. Consolidating product spectrum and gammatone filterbank for robust speaker verification under noisy conditions. In *International Conference on Intelligent Systems Design and Applications (ISDA)*. IEEE, 2015.

[59] L Ferrer, M McLaren, N Scheffer, Y Lei, M Graciarena, and V Mitra. A noise-robust system for NIST 2012 speaker recognition evaluation. Technical report, SRI International, 2013.

[60] W Fisher, G Doddington, and K Goudie-Marshall. The DARPA speech recognition research database: specifications and status. In *Proc. DARPA Workshop on speech recognition*, pages 93–99, 1986.

[61] David Fleming, Bruno L Giordano, Roberto Caldara, and Pascal Belin. A language-familiarity effect for speaker discrimination without comprehension. *Proceedings of the National Academy of Sciences*, 2014.

[62] Laura Fernández Gallardo, Michael Wagner, and Sebastian Möller. Spectral sub-band analysis of speaker verification employing narrowband and wideband speech. Citeseer.

[63] D Garcia-Romero and C Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems.

[64] Andrew Gibiansky, Sercan Arik, Gregory Diamos, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural text-to-speech. In *NIPS*, 2017.

[65] J Gudnason and M Brookes. Voice source cepstrum coefficients for speaker identification. In *ICASSP*. IEEE, 2008.

[66] J Guo, R Yang, H Arsikere, and A Alwan. Robust speaker identification via fusion of subglottal resonances and cepstral features. *The Journal of the Acoustical Society of America*, 141(4):EL420–EL426, 2017.

[67] Mazin Hamad and Mustafa Hussain. *Mazin's Thesis (064055)*. 08 2016.

[68] J Hansen and T Hasan. Speaker recognition by machines and humans: A tutorial review. *IEEE Signal processing magazine*, 32(6):74–99, 2015.

[69] J Hansen and V Varadarajan. Analysis and compensation of Lombard speech across noise type and levels with application to in-set/out-of-set speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(2):366–378, 2009.

[70] John HL Hansen, Marigona Bokshi, and Soheil Khorram. Speech variability: A cross-language study on acoustic variations of speaking versus untrained singing. *The Journal of the Acoustical Society of America*, 148(2):829–844, 2020.

[71] Michael HL Hecker. *Speaker recognition: An interpretive survey of the literature*. American Speech and Hearing Association Washington DC, 1971.

[72] H Hermansky. Perceptual linear predictive (plp) analysis of speech. *The Journal of the Acoustical Society of America*, 87, 1990.

[73] James Hillenbrand, Laura A Getty, Michael J Clark, and Kimberlee Wheeler. Acoustic characteristics of american english vowels. *The Journal of the Acoustical society of America*, 97, 1995.

[74] H Hirsch and D Pearce. The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *ISCA Tutorial and Research Workshop*, 2000.

[75] Gary L Holmgren. Physical and psychological correlates of speaker recognition. *Journal of Speech and Hearing Research*, 10(1):57–66, 1967.

[76] Helge Homburg, Ingo Mierswa, Bülent Möller, Katharina Morik, and Michael Wurst. A benchmark dataset for audio classification and clustering. In *International Society for Music Information Retrieval Conference*, 2005.

[77] X Huang, A Acero, and H Hon. *Spoken language processing: A guide to theory, algorithm, and system development*, volume 95.

[78] Taisuke Ito, Kazuya Takeda, and Fumitada Itakura. Analysis and recognition of whispered speech. *Speech Communication*, 45(2):139–152, 2004.

[79] Anil K Jain, Arun Ross, and Salil Prabhakar. An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, 14(1):4–20, 2004.

[80] C Jankowski, A Kalyanswamy, S Basson, and J Spitz. Ntimit: A phonetically balanced, continuous speech, telephone bandwidth speech database. In *IEEE ICASSP*, pages 109–112, 1990.

[81] Corentin Jemine et al. Master thesis: Automatic multispeaker voice cloning. *Universite de Liege, Liege, Belgique*, 2019.

[82] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *NeurIPS*, 2018.

[83] Jee-weon Jung, Hee-Soo Heo, Ju-ho Kim, Hye-jin Shim, and Ha-Jin Yu. Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification. *arXiv:1904.08104*, 2019.

[84] Jee-weon Jung, Seung-bin Kim, Hye-jin Shim, Ju-ho Kim, and Ha-Jin Yu. Improved rawnet with feature map scaling for text-independent speaker verification using raw waveforms. *Proc. Interspeech 2020*, 2020.

[85] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. In *ICML*, 2018.

[86] Yuta Kawakami, Longbiao Wang, Atsuhiko Kai, and Seiichi Nakagawa. Speaker identification by combining various vocal tract and vocal source features. In *Text, Speech and Dialogue*. Springer, 2014.

[87] C Kim and R Stern. Power-normalized cepstral coefficients (PNCC) for robust speech recognition. In *ICASSP*. IEEE, 2012.

[88] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[89] T Kinnunen. Long-term F0 modeling for text-independent speaker recognition.

[90] Tomi Kinnunen. Spectral features for automatic text-independent speaker recognition. *Licentiate's thesis*, 2003.

[91] Tomi Kinnunen and Haizhou Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech communication*, 52(1):12–40, 2010.

[92] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *arXiv preprint arXiv:1706.02515*, 2017.

[93] N Krishnamurthy and J Hansen. Babble noise: modeling, analysis, and applications. *IEEE TASLP*, 2009.

[94] Edith Law, Kris West, Michael I Mandel, Mert Bay, and J Stephen Downie. Evaluation of algorithms using games: The case of music tagging. In *International Society for Music Information Retrieval Conference*, 2009.

[95] L Li, D Wang, A Rozi, and T Zheng. Cross-lingual speaker verification with deep feature learning. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. IEEE, 2017.

[96] Lantian Li and Thomas Fang Zheng. Gender-dependent feature extraction for speaker recognition. In *IEEE China Summit and International Conference on Signal and Information Processing*. IEEE, 2015.

[97] Qi Li, Jinsong Zheng, Augustine Tsai, and Qiru Zhou. Robust endpoint detection and energy normalization for real-time speech and speaker recognition. *IEEE Transactions on Speech and Audio Processing*, 10(3):146–157, 2002.

[98] X Li, F Li, X Fern, and R Raich. Filter shaping for convolutional neural networks. In *ICLR*, 2017.

[99] Liang Lu, Yuan Dong, Xianyu Zhao, Jiqing Liu, and Haila Wang. The effect of language factors for robust speaker recognition. In *IEEE ICASSP*. IEEE, 2009.

[100] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *JMLR*, 2008.

[101] Richard J Mammone, Xiaoyu Zhang, and Ravi P Ramachandran. Robust speaker recognition: A feature-based approach. *IEEE signal processing magazine*, 13(5):58, 1996.

[102] L Mary and B Yegnanarayana. Extraction and representation of prosodic features for language and speaker recognition. *Speech Communication*, 50, 2008.

[103] Leena Mary and B Yegnanarayana. Prosodic features for speaker verification. In *ICSLP*, 2006.

[104] JS Mason and J Thompson. Gender effects in speaker recognition. *ICSP*, 1993.

[105] P Matějka, O Glembek, O Novotnỳ, O Plchot, F Grézl, L Burget, and J Cernockỳ. Analysis of dnn approaches to speaker identification. In *IEEE ICASSP*, pages 5100–5104, 2016.

[106] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler. OMRAS2 metadata project 2009. In *International Society for Music Information Retrieval Conference*, 2009.

[107] T May, S Van De Par, and A Kohlrausch. Noise-robust speaker recognition combining missing data techniques and universal background modeling. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):108–121, 2012.

[108] Frances McGehee. The reliability of the identification of the human voice. *The Journal of General Psychology*, 17(2):249–271, 1937.

[109] Cory Mckay. A large publicly accessible prototype audio database for music research. In *International Society for Music Information Retrieval Conference*, 2006.

[110] M McLaren, L Ferrer, D Castan, and A Lawson. The 2016 speakers in the wild speaker recognition evaluation. In *INTERSPEECH*, pages 823–827, 2016.

[111] Mahnoosh Mehrabani and John Hansen. Speaker clustering for a mixture of singing and reading. *INTERSPEECH*, 2012.

[112] Mahnoosh Mehrabani and John HL Hansen. Singing speaker clustering based on subspace learning in the GMM mean supervector space. *Speech Communication*, 2013.

[113] B Milner and X Shao. Speech reconstruction from mel-frequency cepstral coefficients using a source-filter model. In *Interspeech*, 2002.

[114] Ji Ming, Timothy J Hazen, James R Glass, and Douglas A Reynolds. Robust speaker recognition in noisy conditions. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1711–1723, 2007.

[115] V Mitra, H Franco, M Graciarena, and Arindam Mandal. Normalized amplitude modulation features for large vocabulary noise-robust speech recognition. In *ICASSP*. IEEE, 2012.

[116] P Mowlaee, R Saeidi, and Y Stylianou. Phase importance in speech processing applications. In *ISCA*, 2014.

[117] Hannah Muckenhirn, Vinayak Abrol, Mathew Magimai Doss, and Sébastien Marcel. Understanding and visualizing raw waveform-based CNNs. In *INTERSPEECH*, 2019.

[118] Hannah Muckenhirn, Mathew Magimai Doss, and Sébastien Marcel. Towards directly modeling raw speech signal for speaker verification using CNNs. In *ICASSP*. IEEE, 2018.

[119] Hannah Muckenhirn, Mathew Magimai-Doss, and Sébastien Marcel. End-to-end convolutional neural network-based voice presentation attack detection. In *IJCB*. IEEE, 2017.

[120] L Muda, M Begam, and I Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *CoRR*, 2010.

[121] K Murty and B Yegnanarayana. Combining evidence from residual phase and MFCC features for speaker recognition. *Signal Processing Letters*, 13, 2006.

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

[123] Sergey Novoselov, Aleksei Gusev, Artem Ivanov, Timur Pekhovsky, Andrey Shulipa, Galina Lavrentyeva, Vladimir Volokhov, and Alexandr Kozlov. Stc speaker recognition systems for the voices from a distance challenge. *arXiv preprint arXiv:1904.06093*, 2019.

[124] Aaron Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[125] Özgür Devrim Orman and Levent M Arslan. Frequency analysis of speaker identification. In *A Speaker Odyssey-The Speaker Recognition Workshop*, 2001.

[126] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an ASR corpus based on public domain audio books. In *ICASSP*. IEEE, 2015.

[127] H Parthasarathi, R Padmanabhan, and H Murthy. Robustness of group delay representations for noisy speech signals. *International Journal of Speech Technology*, 14, 2011.

[128] A Paszke, S Gross, S Chintala, and G Chanan. Pytorch, 2017.

[129] H. A. Patil, M. C. Madhavi, and N. H. Chhayani. Person recognition using humming, singing and speech. In *International Conference on Asian Language Processing*, 2012.

[130] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. 2011.

[131] Sandra Pruzansky. Pattern-matching procedure for automatic talker recognition. *The Journal of the Acoustical Society of America*, 35(3):354–358, 1963.

[132] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet. In *Spoken Language Technology Workshop (SLT)*. IEEE, 2018.

[133] D Reynolds. *A Gaussian mixture modeling approach to text-independent speaker identification*. PhD thesis, Georgia Institute of Technology, 1992.

[134] D Reynolds. Experimental evaluation of features for robust speaker identification. *IEEE Transactions on Speech and Audio Processing*, 2, 1994.

[135] D Reynolds and R Rose. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE TSAP*, 3(1):72–83, 1995.

[136] Douglas A Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech communication*, 17(1-2):91–108, 1995.

[137] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

[138] F Richardson, M Brandstein, J Melot, and D Reynolds. Speaker recognition using real vs synthetic parallel data for dnn channel compensation. *Interspeech*, pages 2796–2800, 2016.

[139] Fred Richardson, Douglas Reynolds, and Najim Dehak. Deep neural network approaches to speaker and language recognition. *IEEE signal processing letters*, 22(10):1671–1675, 2015.

[140] Arun Ross, Sudipta Banerjee, and Anurag Chowdhury. Security in smart cities: A brief review of digital forensic schemes for biometric data. *Pattern Recognition Letters*, 2020.

[141] S Sadjadi and J Hansen. Hilbert envelope based features for robust speaker identification under reverberant mismatched conditions. In *ICASSP*. IEEE, 2011.

[142] S Sadjadi and J Hansen. Robust front-end processing for speaker identification over extremely degraded communication channels. In *IEEE ICASSP*, pages 7214–7218, 2013.

[143] S Sadjadi, M Slaney, and L Heck. MSR identity toolbox v1.0: A MATLAB toolbox for speaker-recognition research. *Speech and Language Processing Technical Committee Newsletter*, 1(4), 2013.

[144] Eugenia San Segundo, Pedro Univaso, and Jorge Gurlekian. Sistema multiparamétrico para la comparación forense de hablantes. *Estudios de fonética experimental*, pages 13–45, 2019.

[145] J Schatzman. Accuracy of the discrete fourier transform and the fast fourier transform. *Journal on Scientific Computing*, 17, 1996.

[146] Markus Schedl, Nicola Orio, Cynthia CS Liem, and Geoffroy Peeters. A professionally annotated and enriched multimodal data set on popular music. In *ACM Multimedia Systems Conference*, 2013.

[147] F Schroff, D Kalenichenko, and J Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[148] Klaus Seyerlehner, Gerhard Widmer, and Peter Knees. Frame level audio similarity-a codebook approach. In *International Conference on Digital Audio Effects*, 2008.

[149] Klaus Seyerlehner, Gerhard Widmer, and Tim Pohle. Fusing block-level features for music similarity estimation. *International Conference on Digital Audio Effects*, 2010.

[150] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. In *ICASSP*. IEEE, 2018.

[151] Elizabeth Shriberg, Luciana Ferrer, Sachin Kajarekar, Anand Venkataraman, and Andreas Stolcke. Modeling prosodic feature sequences for speaker recognition. *Speech Communication*, 46(3-4):455–472, 2005.

[152] Elizabeth Shriberg, Martin Graciarena, Harry Bratt, Andreas Kathol, Sachin S Kajarekar, Huda Jameel, Colleen Richey, and Fred Goodman. Effects of vocal effort and speaking style on text-independent speaker verification. In *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[153] Rita Singh. *Profiling humans from their voice*. Springer, 2019.

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

[155] Alex Solomonoff, William M Campbell, and Ian Boardman. Advances in channel compensation for svm speaker recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–629. IEEE, 2005.

[156] Alex Solomonoff, Carl Quillen, and William M Campbell. Channel compensation for svm speaker recognition. In *Odyssey*, volume 4, pages 219–226. Citeseer, 2004.

[157] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv:1412.6806*, 2014.

[158] R.E Stone, Thomas Cleveland, Johan Sundberg, and Jan Prokop. Aerodynamic and acoustical measures of speech, operatic, and broadway vocal styles in a professional female singer. *Journal of Voice*, 2003.

[159] Catherine Stupp. Fraudsters used AI to mimic CEO's voice in unusual cybercrime case. *The Wall Street Journal*, 30, 2019.

[160] Arend Sulter, Harm Schutte, and Donald Miller. Differences in phonetogram features between male and female subjects with and without vocal training. *Journal of Voice*, 1996.

[161] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2058–2065, 2016.

[162] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[163] Johan Sundberg. The acoustics of the singing voice. *Scientific American*, 1977.

[164] Derek Tingle, Youngmoo E Kim, and Douglas Turnbull. Exploring automatic music annotation with "acoustically-objective" tags. In *International Conference on Multimedia Information Retrieval*, 2010.

[165] A Varga and H Steeneken. Assessment for automatic speech recognition: II. NOISEX-92: a database and an experiment to study the effect of additive noise on speech recognition systems. *Speech communication*, 12(3):247–251, 1993.

[166] A Varga and JM Steeneken. Assessment for automatic speech recognition: Ii. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Communication*, 12, 1993.

[167] N Wang, P Ching, N Zheng, and T Lee. Robust speaker recognition using both vocal source and vocal tract features estimated from noisy input utterances. In *IEEE International Symposium on Signal Processing and Information Technology*, pages 772–777, 2007.

[168] Yuxuan Wang et al. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *ICML*, 2018.

[169] Graham Welch, Desmond Sergeant, and F MACCURTAIN. Some physical characteristics of the male falsetto voice. *Journal of Voice*, 2:151–163, 12 1988.

[170] Jared J Wolf. Acoustic measurements for speaker recognition. *The Journal of the Acoustical Society of America*, 46(1A):89–90, 1969.

[171] Wei Wu, Thomas Fang Zheng, Ming-Xing Xu, and Huan-Jun Bao. Study on speaker verification on emotional speech. In *ICSLP*, 2006.

[172] Hossein Zeinali, Kong Aik Lee, Jahangir Alam, and Lukas Burget. Short-duration speaker verification (sdsv) challenge 2020: the challenge evaluation plan. *arXiv preprint arXiv:1912.06311*, 2019.

[173] Chi Zhang and John HL Hansen. Whisper-island detection based on unsupervised segmentation with entropy-based speech feature processing. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):883–894, 2010.

[174] X Zhang and Y LeCun. Text understanding from scratch. *CoRR*, abs/1502.01710, 2015.

[175] Z Zhang, L Wang, A Kai, T Yamada, W Li, and M Iwahashi. Deep neural network-based bottleneck feature and denoising autoencoder-based dereverberation for distant-talking speaker identification. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):12, 2015.

[176] X Zhao and D Wang. Analyzing noise robustness of MFCC and GFCC features in speaker identification. In *ICASSP*. IEEE, 2013.

[177] X Zhao, Y Wang, and D Wang. Robust speaker identification in noisy and reverberant conditions. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(4):836–845, 2014.

[178] N Zheng, T Lee, and P Ching. Integration of complementary acoustic features for speaker recognition. *Signal Processing Letters*, 14, 2007.

[179] Y Zhou and Z Zhao. Fast ICA for multi-speaker recognition system. In *International Conference on Intelligent Computing*. Springer, 2010.