DeepVOX: Discovering Features from Raw Audio for Speaker Recognition in Degraded Audio Signals

Anurag Chowdhury, Student Member, IEEE, Arun Ross, Senior Member, IEEE

Abstract—Automatic speaker recognition algorithms typically use predefined 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 degradations in the input audio. 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 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.

Index Terms—Speaker Recognition, Degraded Audio, Deep Learning, Feature Extraction, Filterbanks

1 INTRODUCTION

A UTOMATIC speaker recognition entails recognizing an individual from their voice. One of the key applications of speaker recognition is securing devices with voice-controlled user interfaces (VUI), such as digital voice assistants [4] and telephone banking systems [7]. VUIs are gaining popularity due to the ease-of-access provided by their hands-free operation. VUIs are also being adopted in consumer applications, such as Apple’s voice-control feature, for improving accessibility for users with physical disabilities [3], thus broadening the utility of speaker recognition.

A typical automatic speaker recognition (ASR) system has a sensor, a feature extractor, and a matcher. The sensor records speech audio, the feature extractor characterizes the speech audio, and the matcher compares speech characteristics from two audio samples to render a match or non-match decision. In practice, the voice input to the sensor is often degraded with background noise and ambient reverberations. The detrimental effect of these audio degradations propagates through different components of the ASR, consequently lowering its performance [11], [55]. While prior knowledge of the type and extent of audio degradation may be used to partly mitigate its negative effects, noise estimation in speech audio is in itself a challenging task [57]. For example, speech audio recorded in a coffee shop might exhibit various types of background noise, such as babble noise from customers and machinery noise from coffee machines. Estimating the noise in this scenario can be extremely challenging due to its highly dynamic nature. Therefore, it is important to develop speaker recognition techniques that are robust to a wide variety of audio degradations, thereby, providing generalizable speaker recognition performance.

Some of the latest ASR-enabled consumer devices address the issues of audio degradation at the sensor-level by employing specialized hardware, such as far-field microphone arrays [5]. But the use of specialized hardware interfaces limit their compatibility with existing ASR systems. On the other hand, some of the latest speaker recognition techniques [10], [11], [62] address the issues of audio degradations at the software-level by designing noise-robust matchers. But these techniques rely on the use of handcrafted speech features such as Mel-Frequency Cepstral Coefficients (MFCC) and Linear Predictive Coding (LPC). The representation capability of such hand-crafted features varies with the quality of input audio [26], [66], thus limiting the effectiveness of the subsequent matcher.

We position our work with the existing literature by approaching the issue of audio degradation at the feature-level. We design a noise-robust speech feature extractor compatible with existing speaker recognition algorithms. Our method delivers generalizable noise-robust speaker recognition performance without any specialized hardware interface or relying on any handcrafted feature extraction techniques. Our main contributions in this work are as follows:

1) We propose a Convolutional Neural Network (CNN) based approach for learning a noise-robust speech filterbank, referred to as DeepVOX, directly from raw speech audio.
2) We propose an adaptive triplet mining technique for training the proposed DeepVOX filterbank in conjunction with 1D-Triplet-CNN [11], a CNN based speech feature embedding technique, to perform speaker verification.
3) We experimentally demonstrate the compatibility and the associated performance benefits of the DeepVOX features with some of the existing speaker recognition algorithms such as the xVector-
PLDA [62] and the iVector-PLDA [16].

4) 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.

5) Finally, we perform a detailed ablation study of the proposed method for identifying the type of speech features extracted by the DeepVOX filterbanks. We further use guided backpropagation on the learned DeepVOX filterbanks to characterize their frequency-response to a variety of degraded speech audio.

In the next section, we discuss the voice features encoded by some popular speech representation techniques such as MFCC [46] and LPC [39]. We also compare these techniques with the proposed DeepVOX technique and discuss their utility in different scenarios.

2 RELATED WORK

Speech recognition—i.e., recognition and translation of spoken language into machine-readable format—has been one of the most popular tasks in the speech processing community for decades. Therefore, most of the initial speech feature representations were developed from the speech recognition perspective. The widely popular Mel-Frequency Cepstral Coefficients (MFCC) was initially proposed for performing monosyllabic word recognition [14] and was later observed to be efficient for performing speaker recognition as well [54]. The ability of MFCC features to encode both speech and speaker information efficiently makes it a very effective speech representation. However, MFCC features are not robust to audio degradations and are, therefore, not very suitable for speaker recognition tasks in presence of noisy speech data. This has motivated the development of robust speech features for performing speaker recognition in noisy audio conditions, as summarized in Table 1.

In the past few decades, specialized speech features have been developed for encoding different physical and acoustic properties of human voice from speech audio. These can be partitioned into several feature categories based upon the type of voice features they encode [33].

• Short-term spectral features, are usually used to encode vocal tract shape of speakers from speech audio.

• Vocal source features, are usually used to characterize the glottal excitation signal.

• Prosodic features, are usually used to model the speaking style of a speaker.

• High-Level Features, are usually used to model the lexicon of a speaker.

According to the source-filter model of speech [40], human vocal tract can be assumed to behave like a time-varying digital filter due to the articulatory movements. Therefore, in order to model the vocal tract, short-term audio frames (usually 25 to 50ms) are used for extracting the stable voice characteristics in the form of short-term spectral features. Majority of the popular techniques [30] for extracting stable voice characteristics are based on either MFCC or Linear Predictive Coding (LPC). The MFCC feature extraction process uses triangular-filters placed on the Mel-scale for modeling the human audition perception system [46]. The LPC, on the other hand, estimates an all-pole model of filter design for modeling the vocal tract [40].

Humans are noted to be efficient in performing speaker recognition in the presence of unknown type of audio degradations, also referred to as speaker recognition in mis-matched noise conditions. However, the MFCC feature, which is based on human auditory processing, is unable to cope well in such scenarios [68]. Motivated by this, the authors in [68] propose the Gammatone Filterbank as an alternative for 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 [19] and is, thus, a preferred alternative in the MFCC feature extraction process. Additionally, it was proposed [68] to replace the logarithmic rectification step prior to the application of the Discrete Cosine Transform (DCT) in MFCC feature extraction with cubic root, as the logarithmic non-linearity used to compress the dynamic range of filterbank energies is not robust to audio degradations. This new proposed feature set was called Gammatone Frequency Cepstral Coefficients (GFCC) [68]. Another work [39] identified the absence of any form of environmental compensation in the feature extraction process to be one of the key reasons for the poor performance of MFCC features. The authors in [39], hence, proposed noise robust speech features called Power Normalized Cepstral Coefficients (PNCC) that incorporated a noise-suppression algorithm based on asymmetric filtering for suppressing the background excitation. Similar to the GFCC feature, PNCC also uses Gammatone Filterbank instead of Mel-filterbank for extracting voice characteristics.

Another drawback of the MFCC feature extraction process is its disregard of phase information in the speech data, as the features are extracted only from the amplitude spectrum. The initial motivation behind disregarding the phase information was based on human auditory system experiments [19], where short-term phase spectrum did not provide enough performance benefits to justify the added computational complexity of extracting phase-based features. However, recent studies [42], [50] have reported comparable and complementary speaker recognition performance of both magnitude-based and phase-based features [47]. One recent work [56] used the Hilbert transform for combining the amplitude and phase information in speech data to generate a noise-robust and unified feature representation called Mean Hilbert Envelope Coefficient (MHEC). Similar to the GFCC and PNCC features, the MHEC features also use the Gammatone Filterbank. The Hilbert envelope of output of the Gammatone Filterbank is used to compute the MHEC features.

LPC-based methods [67] in comparison, attempt to characterize the speech production model using an all-pole filter model. Linear Prediction Cepstral Coefficients (LPCC) are the cepstral representation of LPC features and are often considered more reliable than the regular LPC features [67]. One of the major disadvantages of the LPC and LPCC-based techniques is that they provide a linear approximation of speech at all frequencies, whereas the spectral resolution of human hearing is known to reduce with frequency beyond 800Hz. This issue was addressed by Hermansky et al. [27] in their work on Perceptual Linear Prediction (PLP) Coefficients. For extracting the PLP features on a non-linear scale, that resembles the human auditory system, several spectral transformations [27] were applied to the power spectrum of the speech audio prior to the all-pole model approximation by the autoregressive model. It is important to note that while LPC and MFCC based features are usually augmented with ‘spectro-temporal features’ in the form of delta and delta-delta coefficients, which are the first and second order time-derivatives of the short-term spectral features, respectively. Spectro-temporal features are one way of adding temporal features, such as formant transitions...
and energy modulations, to the short-term spectral features.

The human vocal tract contributes to a majority of the speaker dependent features in the human voice. Short-term spectral features, such as LPC, that attempt to model the human vocal tract are particularly effective in performing speaker recognition. However, vocal tract modeling is not the only way of approaching speaker recognition. Vocal source features [33] can also be used for the task. Vocal source features refer to the characteristics of the source of human voice originating in the form of glottal excitation pulses. Features such as glottal pulse shape, rate of vocal fold vibration, fundamental frequency, degree of vocal fold opening and the duration of the closing phase can potentially be extracted [18] to characterize vocal source features for performing speaker recognition. One such work in [69] used the inverse vocal tract filter learned using LPC for estimating the LP residual signal (source signal) from the original speech waveform. It further uses a wavelet transform on the Linear Prediction (LP) residual signal for extracting vocal source features called Wavelet Octave Coefficients Of Residues (WOCOR). In their experiments, the authors [69] also showed benefits of supplementing MFCC features with WOCOR features for improving overall speaker verification performance.

Unlike some other biometric modalities, such as fingerprint and face, human voice as a biometric modality is a combination of both physical and behavioral traits. While vocal tract and vocal source features capture the physical aspects of the human voice production system, the behavioral aspects are captured by prosodic and high-level features. Prosodic features capture non-segmental aspects of speech such as the intonation, speaking style, accent, and pronunciation of the speaker. Unlike short-term spectral features, prosodic features are extracted from longer segments of speech and are also less sensitive to channel effects [15]. One of the most important prosodic feature is the rate of vibration of the vocal folds during voiced speech generation also known as fundamental frequency or F0. The authors in [31] successfully modeled F0 features, both parametrically and non-parametrically, and combined it with MFCC features to improve speaker verification accuracy in degraded audio signals, proving the effectiveness of prosodic features when supplemented with short-term spectral features in degraded audio conditions. In other literature [15], [20], the authors used Joint Factor Analysis for modeling prosody from speech data and then fused it with MFCC features for improving speaker verification performance. In [17], the author extracted high level speech features, such as word unigrams and bigrams, for modeling speaker identities based on idiolecal differences between different speakers. The authors were able to encode the lexicon of speakers from their speech data to derive a relationship between speaker identity and their usage of different words in their spoken language.

In the past decade, deep learning based methods have been successfully designed and implemented for solving many speech processing tasks, including speaker recognition [10], [11], [36]. A majority of such speaker recognition methods use some type of hand-crafted features, e.g. MFCC, LPC, as input to their network for solving the problem. For example, authors in [36] developed an end-to-end Neural Speaker Embedding System called Deep Speaker that learns speaker-specific embeddings from 64-dimensional log Mel-filterbank coefficients using ResCNN and GRU architectures. However, some of the recent works [44], [45], [53] have proposed to feed the raw speech waveform directly as input to deep neural networks for performing a variety of tasks such as speech recognition, speaker recognition and even to detect voice presentation attacks. The authors in [53], for example, propose to learn the cut-off frequency of pre-defined band-pass filters for performing speaker recognition on clean (un-degraded) speech data.

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 cues from our previous work on 1D-CNN [10] and 1D-Triplet-CNN [11] based architectures for performing speaker identification and verification respectively from degraded audio signals. However, both these architectures use MFCC and LPC-based feature representation as input and are, therefore, limited by the representation power of MFCC and LPC features. We, instead, propose a 1D-CNN based feature extraction module, termed as DeepVOX, to learn and extract speech feature representation directly from raw audio data, in the time-domain itself. The DeepVOX learns filterbanks directly from a large quantity of degraded raw speech audio samples, thereby laying its emphasis on learning highly discriminative speech audio features robust to audio degradations.

Note that, unlike the work in [53], 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 [44], [45], [53], we demonstrate the compatibility of the proposed DeepVOX features with some state-of-the-art deep learning-based speaker recognition

| Paper | Feature Category | Feature Details | Comments |
|-------|------------------|----------------|---------|
| Davis and Mermelstein [14] | Mel-Frequency Cepstral Coefficients (MFCC) | | |
| Zhao et al. [69] | Gammatone Frequency Cepstral Coefficients (GFCC) | | |
| Mammone et al. [59] | Short-term spectral features | Linear Predictive Cepstral Coefficients (LPCC) | Useful for modeling vocal tract shape |
| Huang et al. [28] | Vocal source features | Perceptual Linear Prediction (PLP) coefficients | |
| Hermansky et al. [27] | Prosodic features | Voice Source Cepstrum Coefficients (VSCC) | Useful for modeling lexicon of a speaker |
| Todisco et al. [64] | High-Level Features | Joint Factor Analysis based Prosody Modeling | |
| Mitrja et al. [14] | Vocal source features | Wavelet Octave Coefficients of Residues (WOCOR) | Useful for characterizing the glottal excitation signal |
| Kim et al. [10] | Vocal source features | Voice Source Cepstrum Coefficients (VSCC) | |
| Nalpadi et al. [56] | Vocal source features | Voice Source Cepstrum Coefficients (VSCC) | |
| Zheng et al. [69] | Vocal source features | Voice Source Cepstrum Coefficients (VSCC) | |
| Guidanoson et al. [15] | Vocal source features | Voice Source Cepstrum Coefficients (VSCC) | |
| Kinnunen [31] | Prosodic features | Logarithmic Fundamental Frequency (F0) features | Useful for modeling speaking style of a speaker |
| Ferrer et al. [20] | Vocal source features | Voice Source Cepstrum Coefficients (VSCC) | |
| Doddington [17] | Vocal source features | Voice Source Cepstrum Coefficients (VSCC) | |

Table 1: Existing speech feature representations used for speaker recognition, as categorized by Kinnunen et al. [33].
methods such as xVectors \cite{62} and 1D-Triplet-CNN \cite{11} and even on classical statistics-based methods such as the iVector-PLDA \cite{16}. The next few sections present our proposed DeepVOX architecture for performing speaker recognition. We also conduct 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.

3 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). 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.

3.1 Short-term Speech Feature Extraction Using DeepVOX

In this work, we use the proposed DeepVOX feature extractor jointly with a 1D-Triplet-CNN \cite{11}-based feature embedding network for performing speaker recognition. The 1D-Triplet-CNN \cite{11} 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 2), also referred to as 1D-Triplet-CNN(DeepVOX), consists of four separate units described below:

3.1.1 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 \cite{6} 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 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 $160 \times 200$. The extracted speech frames are then made into speech frame triplets for inputting into the proposed DeepVOX architecture.

3.1.2 Speech Frame Triplets

The authors in \cite{60} introduced the idea of triplet based CNNs. As illustrated in Figure 2, 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 $X \neq Y$. 

![Figure 1. A visual representation of the proposed Dilated 1D-CNN based DeepVOX feature extraction process.](image-url)
3.1.3 DeepVOX

The DeepVOX architecture, as given in Figure 2, 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 \( \text{[11]} \) and SELU \( \text{[34]} \) (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.

3.1.4 1D-Triplet-CNN

The architecture of 1D-Triplet-CNN, similar to the architecture of DeepVOX, 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 [48], where the authors use dilated convolutions to increase the receptive field size non-linearly 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 \( \text{[11]} \) 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 2). 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.

3.1.5 Cosine Triplet Embedding Loss

The cosine triplet embedding loss \( \text{[11]} \) is a modification upon the triplet loss initially introduced in [60] by replacing the euclidean distance metric with cosine similarity. As noted in \( \text{[11]} \), using cosine similarity leads to a faster convergence and more stable learning due to its bounded nature. The triplet loss \( \text{[60]} \) is designed to learn an embedding \( g(x) \in \mathbb{R}^d \), where \( f(x) \) is DeepVOX feature of speech frame \( x \) and \( g(x) \) is its embedding in a \( d \)-dimensional euclidean space \( \mathbb{R}^d \). In this work, \( d \) is set to 128. The embedding is learned in such a fashion 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 \( \text{[11]} \), is given by:

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

Here, \( L(\cdot, \cdot, \cdot) \) is the cosine triplet embedding loss function. \( S_a \), the anchor sample, is a speech sample from a subject ‘X’. \( S_p \), the positive sample, is another speech sample from the same subject ‘X’. \( S_n \), the negative sample, is a speech sample from another subject ‘Y’, such that \( X \neq Y \). \( \alpha_{\text{margin}} \) 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 3.1.4, 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.
Figure 3. A visual representation of adaptive triplet mining used to train the DeepVOX architecture using 1D-Triplet-CNN.

As illustrated in Figure 3, 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 4) 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 \( \text{Score}_{match}(S_1, S_2) \) for the given speech pair is computed using the cosine similarity metric as follows:

\[
\text{Score}_{match}(S_1, S_2) = \cos(g(f(S_1, S_2)))
\]

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

3.1.6 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 [11], depends on the difficulty of the triplets generated from the training data. The authors in [11] 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 techniques use either hard or semi-hard triplet mining [60], we propose an alternative adaptive triplet mining technique.

In adaptive triplet mining, at a given epoch \( i \), the goal is to select a negative sample \( S^i_n \), such that:

\[
\cos(g(f(S^i_a, S^i_p))) > \cos(g(f(S^i_a, S^i_n))) + \alpha_{margin}
\]

\[
\tau_{S^i_n} > \tau_{S^i_n-1}
\]

Where, \( S^i_a \) is the anchor speech sample, \( S^i_p \) is the positive speech sample, and \( \alpha_{margin} \) is the margin, as also illustrated in Figure 3. Here, \( \tau_{S^i_n} \) is a parameter that denotes the average difficulty of \( S^i_n \) (a negative sample), chosen at epoch \( i \). A value of \( \tau = 0 \) yields the easiest negative sample and \( \tau = 1 \) yields the hardest negative sample, as shown in Figure 3. In our experiments, the value of \( \tau \) is determined by the current stage (or epoch) of the training process. We initialize the training process with the value of \( \tau \) at 0.4 (empirically chosen) and increase it gradually to 1.0 through the course of the training process. This is done to ensure a minimum difficulty of the training triplets at the beginning of the training process 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 process [60]. 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, to ensure easier initialization of the training process we perform softmax pre-training of the model in the identification mode.

3.2 Analysis of the Proposed DeepVOX Architecture

In Section 3.1.3 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. 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. 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.

3.2.1 Building Blocks of Short-term Spectral Feature Extraction Algorithms

The comparison in Figure 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 a 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. 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. Mel frequency-band 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: The nonlinear rectification step is done to compress the dynamic range of filterbank energies. The importance of this step is demonstrated in 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.

### 3.2.2 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, 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 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 fourier domain. Fourier Transform is usually implemented using the Fast Fourier Transform (FFT) algorithm which makes the filtering of 1D audio signals even more computationally efficient, \( O(n \log n) \), as compared to performing general convolution operation, \( O(n^2) \). However, FFT only provides a close approximation of time domain filtering and is often inconsistent across different implementations of the FFT algorithm, 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 Convolutional Neural Networks (CNN) in our algorithm to learn time-domain filters efficiently from raw speech audio.

As discussed earlier and illustrated in Figures 1 and 2 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:

\[
S = [u_1, u_2, \ldots, u_i, \ldots, u_n]
\]

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:

\[
\mathbf{O} = [f_1, f_2, \ldots, f_i, \ldots, f_n]
\]

Where, \( f_i \) is given by:

\[
f_i = \begin{bmatrix}
x_{i,1} \\
x_{i,2} \\
\vdots \\
x_{i,j} \\
\vdots \\
x_{i,40}
\end{bmatrix}
\]

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:

\[
f_i = (l_m \ast (l_{m-1} \ast \cdots l_k \ast l_1(u_i)))
\]  \hspace{1cm} (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:

\[
l_k(u_i) = C_k \ast u_i \text{,}
\]  \hspace{1cm} (9)

where \( C_k \) is the convolutional filter for the \( k^{th} \) layer. The operation in the Equation 9 is equivalent to time-domain filtering of input signal \( u_i \) with filter \( C_k \). Hence, we can rewrite equation 8 as:

\[
f_i = (C_m \ast (C_{m-1} \ast \cdots C_k \ast (\cdots C_1 \ast (u_i)))) \text{,}
\]  \hspace{1cm} (10)

Since, the convolution operation is associative, we can rewrite equation 10 as:

\[
f_i = (C_m \ast C_{m-1} \ast \cdots C_k \ast C_1) \ast u_i
\]  \hspace{1cm} (11)

DeepVOX filterbank = \( C_m \ast C_{m-1} \ast \cdots C_k \ast C_1 \text{.} \)  \hspace{1cm} (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.

In the following sections, we will discuss the various datasets used and experiments performed for evaluating the performance of our proposed algorithm.

4 Datasets and Experiments

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

1) Fisher English Training Speech Part 1 dataset \[\text{[13]}\]
2) NIST SRE 2008 and 2010 datasets \[\text{[1]}\]

We also create degraded versions of the above speech datasets by adding different types of noise data from the NOISEX-92 \[\text{[65]}\] 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 proposed method to a wide variety of audio degradations. Additionally, all the speech datasets were sampled at a rate of 8,000Hz to match the NIST SRE dataset specifications \[\text{[1]}\]. We also perform speaker verification experiment on speech samples of varying audio lengths, as also done in \[\text{[11]}\]. 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.

4.1 Datasets

4.1.1 VOXCeleb2 Dataset

The VoxCeleb2 \[\text{[12]}\] 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 1,45,569 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 4.2 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.

4.1.2 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 \[\text{[65]}\] 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 \[\text{[65]}\] 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 3 and Figure 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 3 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 3 are termed as cross-noise experiments, since the training and testing sets are degraded with different types of noise.

4.1.3 NIST SRE 2008 and 2010 Datasets

The NIST SRE 2008 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 and Figure 5, is to evaluate the performance of our proposed algorithm in the presence of multi-lingual data, as cross-lingual speaker recognition \[\text{[37]}\] 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 NIST SRE 2010 [2] dataset.

### 4.2 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 2, 3, 4 and 6. Since the xVector implementation requires a DeepVOX model to be trained. For each of the experiments in Tables 2 and 4, we use speech data only from corresponding training set to train the DeepVOX model, as discussed in Section 3.1 ensuring disjoint data and subjects in the training and testing sets for the DeepVOX feature extraction process.

- **iVector-PLDA-based Speaker Verification Experiments:** We conduct experiments using iVector-PLDA [24] 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 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 [58] implementation of the iVector-PLDA algorithm for conducting our experiments.

- **xVector-PLDA-based Speaker Verification Experiments:** We also use the xVector-PLDA [62] algorithm to establish a neural network-based baseline performance for the experiments reported in Tables 2, 3 and 4. Since the xVector implementation in the Kaldi toolkit only supports 24-dimensional MFCC features, we use the PyTorch-based implementation of the xVector algorithms trained on the baseline features (MFCC, LPC, MFCC-LPC) and DeepVOX features separately, to evaluate and compare the effectiveness of DeepVOX features, with respect to baseline features, in classical speaker recognition algorithms. However, unlike the baseline features, DeepVOX feature extraction process requires a DeepVOX model to be trained. For each of the experiments in Tables 2, 3, and 4 we use speech data only from corresponding training set to train the DeepVOX model, as discussed in Section 3.1 ensuring disjoint data and subjects in the training and testing sets for the DeepVOX feature extraction process.

### 4.2.1 Baseline Speaker Verification Experiments

For establishing baseline speaker verification performance on the VOXCeleb2, Fisher, and NIST SRE 2008 and 2010 speech datasets, we choose iVector-PLDA [24] and xVector-PLDA [62] algorithms trained on the baseline features (MFCC, LPC, MFCC-LPC) and DeepVOX features separately, to evaluate and compare the effectiveness of DeepVOX features, with respect to baseline features, in classical speaker recognition algorithms. However, unlike the baseline features, DeepVOX feature extraction process requires a DeepVOX model to be trained. For each of the experiments in Tables 2, 3, and 4 we use speech data only from corresponding training set to train the DeepVOX model, as discussed in Section 3.1 ensuring disjoint data and subjects in the training and testing sets for the DeepVOX feature extraction process.

### Table 2

| # | Method | TMR@FMR= (EER(in %)) | manDCF (cmiss=1.010) | EER(in %) |
|---|--------|-----------------------|---------------------|----------|
| 1 | 1D-Triplet-CNN-online | 90.49, 96.09 | 0.51, 3.20 | 57.28 |
| 2 | 1D-Triplet-CNN | 79.12, 94.09 | 0.47, 2.63 | 46.22 |
| 3 | iVector-PLDA | 86.16, 90.02 | 0.33, 1.67 | 37.87 |

### Table 3

| # | Train set | Method | TMR@FMR= (EER(in %)) | manDCF (cmiss=1.010) | EER(in %) |
|---|----------|--------|-----------------------|---------------------|----------|
| 1 | F1/F1 | 94.13, 92.06 | 0.88, 5.91 | 0.52, 3.30 | 7.99 |
| 2 | F1/F2 | 79.12, 94.09 | 0.91, 4.60 | 0.51, 3.20 | 33.42 |
| 3 | F3/F1 | 86.16, 90.02 | 0.33, 1.67 | 0.51, 3.20 | 57.28 |
| 4 | F2/F2 | 90.49, 96.09 | 0.51, 3.20 | 57.28 |
| 5 | F2/F1 | 86.16, 90.02 | 0.33, 1.67 | 0.51, 3.20 | 57.28 |

### Table 4

| # | Train set | Method | TMR@FMR= (EER(in %)) | manDCF (cmiss=1.010) | EER(in %) |
|---|----------|--------|-----------------------|---------------------|----------|
| 1 | 1D-Triplet-CNN-online | 90.49, 96.09 | 0.51, 3.20 | 57.28 |
| 2 | 1D-Triplet-CNN | 79.12, 94.09 | 0.47, 2.63 | 46.22 |
| 3 | iVector-PLDA | 86.16, 90.02 | 0.33, 1.67 | 37.87 |
Figure 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 and 2010 datasets (Exp. 6 to 11), and the multilingual subset of NIST SRE 2008 dataset (Exp. 12 to 14) using iVector-PLDA, xVector-PLDA and 1D-Triplet-CNN and 1D-Triplet-CNN-online algorithms on MFCC, LPC, MFCC-LPC, and DeepVOX feature sets.

Figure 6. (a) TMR@FMR=10%, (b) TMR@FMR=1%, (c) \text{minDCF}(\text{C}_{\text{miss}} = 10), and (d) 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.
4.2.3 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 2 and 4) 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 [11] 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 1D-Triplet-CNN embeddings. These pair of embeddings are then compared using the cosine similarity metric to generate a match score.

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

We also perform speaker recognition experiments using the 1D-Triplet-CNN [11] algorithm. These experiments provide benchmark results (given in Tables 3 and 4) 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 features 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 learn both the DeepVOX-based feature representation and the 1D-Triplet-CNN-based speech feature embedding simultaneously. For evaluation, speech audio pairs are fed to the trained 1D-Triplet-CNN model to extract pairs of DeepVOX features which are then fed into the trained 1D-Triplet-CNN model to extract pairs of 1D-Triplet-CNN embeddings. These pair of embeddings are then compared using the cosine similarity metric to generate a match score.
1D-Triplet-CNN embeddings. These pair of embeddings are then compared using the cosine similarity metric to perform matching.

### 4.2.4 1D-Triplet-CNN-based Speaker Recognition Systems 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 2 and 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 3.1, the difficulty 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 reliability of extracted speaker-dependent features in speech data. In experiment 12, the trained models are evaluated on same-language (English Only) trials from a subset of 59 multi-lingual subjects in the testing set of NIST SRE 2008 dataset. This experiment serves to establish the baseline same-language (English to English) speaker verification performance of all the algorithms.

### 4.2.5 Experiments for Studying the Effect of Language on Speaker Verification Language

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 [22, 38]. 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 [38]. In this work, we perform additional speaker recognition experiments (given by experiments 12 to 14 in Table 3) 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 the training set of NIST SRE 2008 dataset. The evaluation sets, however, in experiments 12 to 14 varied, as given below:

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

**Multi-lingual trials:** In experiment 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 the testing set of NIST SRE 2008 dataset. 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 experiment 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 the testing set of NIST SRE 2008 dataset. 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.

### 4.2.6 Speaker Verification Experiments on Audio Samples of Varying Length

The reliability of extracted speaker-dependent features in speech audio depends on the amount of usable speech data in an audio sample, which in turn is directly dependent on the length of the audio sample. In the experiments, the models are trained on English speech data spoken by a subset of 1076 English-speaking subjects in the training set of NIST SRE 2008 dataset. The evaluation sets, however, in experiments 12 to 14 varied, as given below:

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

**Multi-lingual trials:** In experiment 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 the testing set of NIST SRE 2008 dataset. 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 experiment 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 the testing set of NIST SRE 2008 dataset. 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.

### 4.2.6 Speaker Verification Experiments on Audio Samples of Varying Length

The reliability of extracted speaker-dependent features in speech audio depends on the amount of usable speech data in an audio sample, which in turn is directly dependent on the length of the audio sample. In the experiments, the models are trained on English speech data spoken by a subset of 1076 English-speaking subjects in the training set of NIST SRE 2008 dataset. The evaluation sets, however, in experiments 12 to 14 varied, as given below:

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

**Multi-lingual trials:** In experiment 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 the testing set of NIST SRE 2008 dataset. 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 experiment 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 the testing set of NIST SRE 2008 dataset. 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.

### 4.2.6 Speaker Verification Experiments on Audio Samples of Varying Length

The reliability of extracted speaker-dependent features in speech audio depends on the amount of usable speech data in an audio sample, which in turn is directly dependent on the length of the audio sample. In the experiments, the models are trained on English speech data spoken by a subset of 1076 English-speaking subjects in the training set of NIST SRE 2008 dataset. The evaluation sets, however, in experiments 12 to 14 varied, as given below:

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

**Multi-lingual trials:** In experiment 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 the testing set of NIST SRE 2008 dataset. 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 experiment 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 the testing set of NIST SRE 2008 dataset. 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.

### 4.2.6 Speaker Verification Experiments on Audio Samples of Varying Length

The reliability of extracted speaker-dependent features in speech audio depends on the amount of usable speech data in an audio sample, which in turn is directly dependent on the length of the audio sample. In the experiments, the models are trained on English speech data spoken by a subset of 1076 English-speaking subjects in the training set of NIST SRE 2008 dataset. The evaluation sets, however, in experiments 12 to 14 varied, as given below:

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

**Multi-lingual trials:** In experiment 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 the testing set of NIST SRE 2008 dataset. 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 experiment 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 the testing set of NIST SRE 2008 dataset. 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.

### 4.2.6 Speaker Verification Experiments on Audio Samples of Varying Length

The reliability of extracted speaker-dependent features in speech audio depends on the amount of usable speech data in an audio sample, which in turn is directly dependent on the length of the audio sample. In the experiments, the models are trained on English speech data spoken by a subset of 1076 English-speaking subjects in the training set of NIST SRE 2008 dataset. The evaluation sets, however, in experiments 12 to 14 varied, as given below:

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

**Multi-lingual trials:** In experiment 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 the testing set of NIST SRE 2008 dataset. 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 experiment 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 the testing set of NIST SRE 2008 dataset. 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.
audio sample. Therefore, performing speaker recognition in audio samples of a small duration is a challenging task. Since in real-life scenarios, probe audio is 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 6 and Figure 5), 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 data is varied between 3.5 seconds and 0.5 seconds in steps of 0.5 seconds.

5 Results and Analysis

The results for all the experiments described in Section 4.2 are given in Tables 2, 3, 4, 5 and Figures 5 and 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%}), which are the performance metrics for comparison of the baseline methods and the proposed method. The minDCF is reported at two different $C_{miss}$ (cost of a missed detection) values of 1 and 10 (minDCF($C_{miss} = \{1, 10\}$)). Additionally, we also determine and report the proportion of test data pairs where the performance of the proposed and baseline algorithms are comparable and also where they out-performed each other, at False Match Rate of 1%.

- Overall, in all the speaker verification experiments given in Tables 2, 3, 4, 5 and 6, 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.
- In the speaker verification experiment on the VOXCeleb2 dataset, given in Table 2 and Figure 3, 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%}) by 9.89%, 0.9%, and minDCF($C_{miss} = \{1, 10\}$) by $0.08, 0.68$.

2.46%, and minDCF($C_{miss} = \{1, 10\}$) by $0.08, 0.68$. For the xVector-PLDA algorithm, the DeepVOX features improve speaker verification performance (TMR@FMR={1%, 10%}) by 14%, 2.94%, reduces the EER by 3.4% and minDCF($C_{miss} = \{1, 10\}$) by $0.15, 1.56$. 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.
- In Experiment 1 given in Table 2 the 1D-Triplet-CNN-online algorithm correctly verified the same 85.90% of the test samples, across all the features. However, the 1D-Triplet-CNN-online algorithm using the DeepVOX features help to correctly verify an additional 6.58% of the test samples over the MFCC features, 5.67% over the LPC features, and 4.21% over the MFCC-LPC features. However, the 1D-Triplet-CNN-online(DeepVOX) method fails to correctly verify 1.02% of the test samples that were correctly verified by all the baseline methods.
- In all the four speaker verification experiments (Experiments 2 to 5) on the degraded Fisher dataset given in Table 3 and Figure 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.
- 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%) by $23.83%, 10.65%$, reduces the EER by 5.98% and minDCF($C_{miss} = \{1, 10\}$) by $0.24, 2.30$. Similarly, for the 1D-Triplet-CNN algorithm, the DeepVOX features improve speaker verification performance (TMR@FMR=1%) over the MFCC-LPC features by $29.69%, 14.45%$, reduces the EER by 7.11% and minDCF($C_{miss} = \{1, 10\}$) by $0.25, 2.89$. 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($C_{miss} = \{1, 10\}$) by $0.06, 1.03$. However, a performance (TMR@FMR=1%) 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%) over the best performing baseline feature (MFCC-LPC) by $23.45%, 9.61%$. It also reduces the EER by 4.69% and minDCF($C_{miss} = \{1, 10\}$) by $0.17, 2.29$.
- On an average across the four speaker verification experiments (Experiments 2 to 5), given in Table 3, the 1D-Triplet-CNN-online algorithm, across all the features, correctly verify the same 67.96% of the test samples. Furthermore, the 1D-Triplet-CNN-online algorithm using the DeepVOX features correctly verify an
additional 14.75% of the test samples over the MFCC features, 16.68% over the LPC features, and 12.77% over the MFCC-LPC features. However, the 1D-Triplet-CNN-online(DeepVOX) method failed to correctly verify 5.86% of the test samples that were correctly verified by all the baseline methods.

- On an average across the six speaker verification experiments (Experiments 6 to 11), 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 17.58%, 10.58%, reduces the EER by 4.38% and minDCF($C_{miss} = \{1, 10\}$) by {0.13, 1.73}. 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 10.33%, 3.52%, reduces the EER by 1.24% and minDCF($C_{miss} = \{1, 1.10\}$) by {0.07, 1.14}. For the xVector-PLDA algorithm, the DeepVOX features improve speaker verification performance (TMR@FMR=1%) over the best performing baseline feature (MFCC-LPC) by 2.74% and reduces the minDCF($C_{miss} = \{1, 10\}$) by {0.01, 0.25}. However, a performance (TMR@FMR=10%) loss of 3.59% and an increase in EER of 2.97% 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 8.03%, 25.52%. It also reduced the EER by 10.98% and minDCF($C_{miss} = \{1, 10\}$) by {0.001, 1.03}.

- On an average, across the six experiments in Table 4 the 1D-Triplet-CNN-online algorithm, across all the features, correctly verify the same 58.57% of the test samples. Furthermore, the 1D-Triplet-CNN-online algorithm using the DeepVOX features correctly verify an additional 12.62%, 13.08%, and 10.72% of the test samples over the MFCC, LPC, and MFCC-LPC features, respectively. However, the 1D-Triplet-CNN-online(DeepVOX) method fails to correctly verify 5.93% of the test samples that were correctly verified by all the baseline methods.

- In the three speaker verification experiments (Experiments 12 to 14, given in Table 5 on multi-lingual 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($C_{miss} = \{1, 10\}$) by {0.21, 2.21}.

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($C_{miss} = \{1, 10\}$) reduces by {0.17, 2.93}. 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($C_{miss} = \{1, 10\}$) reduces by {0.16, 1.98}. 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($C_{miss} = \{1, 10\}$) reduces by {0.12, 1.76}.

- It is interesting to note the effect of language on verification performance in the Experiments 12 to 14. Best speaker veri-
Figure 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.

- In the experimental results given in Table 6 and illustrated in Figure 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 verification performance is achieved in Experiment 12 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 13) reduces the verification performance (TMR@FMR={1%, 10%}) of 1D-Triplet-CNN-online algorithm 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 14 shows the largest reduction in verification performance, verifying the impact of language-familiarity effect [22], [38] in all the algorithms and features evaluated in our experiments. However, 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%)) in case of the DeepVOX features compared to a reduction of 40.76% for the MFCC-LPC features, 39.31% for the MFCC features, and 37.91% for the LPC features using the best performing algorithm (1D-Triplet-CNN-online).

- In the experimental results given in Table 6 and illustrated in Figure 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 perfor-
formance 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 suffers a performance (TMR@FMR=10%) reduction of 40%, compared to a reduction of 54% using MFCC-LPC features, 66% using MFCC features, and 56% using LPC features, when the audio length is reduced from 6.6 to 0.5 seconds. Similar trends were observed for the 1D-Triplet-CNN algorithm where a performance loss of 40%, 47%, 64%, and 55% is observed for the DeepVOX, MFCC-LPC, MFCC, and LPC features respectively. For the xVector-PLDA algorithm a performance loss of 77%, 85%, 83%, and 82% is observed for the DeepVOX, MFCC-LPC, MFCC, and LPC features respectively. Finally, for the iVector-PLDA algorithm a performance loss of 66%, 81%, 83%, and 90% is observed for the DeepVOX, MFCC-LPC, MFCC, and LPC features respectively. It is important to note that, compared to the 1D-Triplet-CNN based algorithms, relatively larger performance losses are observed for the iVector-PLDA and xVector-PLDA algorithms, across all the features, as also observed in [11]. 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.

- The extraction of speaker dependent features directly from raw audio using the proposed DeepVOX model vastly improves the verification performance compared to the baseline features. This suggests that the proposed DeepVOX feature, compared to the classical hand-crafted features such as MFCC and LPC, is better at extracting short-term speaker dependent speech characteristics from speech audio in presence of audio degradations.

6 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 [43], 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’ [63]. Such an analysis will help us understand the components of a speech 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 [21] dataset due to the availability of ground-truth information for analysis of frequency sub-bands essential for speaker recognition in the TIMIT dataset [23], [32], [49]. 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 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 7, we extract speech utterances corresponding to five phonemes /awl/, /ehl/, /iyl/, /owl/, /awl/ 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 [8] toolkit, as illustrated in Figure 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 [29].

Operational Frequency-range of the DeepVOX Model:

Similar to [43], we represent the input audio signal and corresponding relevance signals on the Power Spectral Density (PSD) plots (given in Figure 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 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 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 [23], [32], [49]. 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: Finally, as shown in Figure 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 [21] dataset, and the additive synthetic car noise from the NOISEX-92 dataset [65]. 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 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.
7 IMPLEMENTATION AND REPRODUCIBILITY

LPC and MFCC features are extracted using the VOICEBOX [9] toolbox. The DeepVOX and 1D-Triplet-CNN models are implemented using PyTorch [51] toolkit and trained on Nvidia TITAN V GPUs. The final version of our DeepVOX implementation will be made publicly available on github.

8 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 [68]. Mel-Frequency bank and Gammatone-frequency bank are two such examples of handcrafted filterbanks used in MFCC and PNCC 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, 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 5.1.1, 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.

ACKNOWLEDGMENT

We would like to thank Dr. Shantanu Chakrabartty and Dr. Kenji Aono from Washington University in St. Louis for providing us their audio degradation tool for degrading the audio samples used in this work. We would also like to thank Dr. Joseph P. Campbell from MIT Lincoln Laboratory for the useful discussions.

REFERENCES

[1] 2008 NIST speaker recognition evaluation training set part 2 LDC2011S07. https://catalog.ldc.upenn.edu/LDC2011S07. Accessed: 2018-03-06.
[2] 2010 NIST speaker recognition evaluation test set LDC2017S06. https://catalog.ldc.upenn.edu/LDC2017S06. Accessed: 2018-03-06.
[3] Apple accessibility. https://www.apple.com/accessibility/iphone/. Accessed: 2020-03-28.
[4] Apple Siri voice recognition. http://www.zdnet.com/article/apple-adds-individual-voice-recognition-to-key-siri-ios-9/ Accessed: 2017-12-29.
[5] Google home voice recognition. https://www.cnet.com/news/is-google-home-good-at-voice-recognition/ Accessed: 2017-12-29.
[6] MATLAB voice activity detection by spectral energy. https://github.com/JarvasChen/MATLAB-Voice-Activity-Detection-by-Spectral-Energy Accessed: 2018-03-06.
[7] Wellsfargo voice verification. https://www.wellsfargo.com/privacy-security/voice-verification/ Accessed: 2017-12-29.
[8] P. Boersma et al. Praat, a system for doing phonetics by computer. Glot international, 5, 2002.
[9] 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.
[10] A. Chowdhury and A. Ross. Extracting sub-glottal and supra-glottal features from MFCC using convolutional neural networks for speaker identification. In ICASSP, IEEE, 2017.
[11] A. Chowdhury and A. 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.
[12] J. S. Chung, A. Nagrani, and A. Zisserman. Voeceleb2: Deep speaker recognition. arXiv preprint arXiv:1806.05622, 2018.
[13] C. Cieri, D. Miller, and K. Walker. Fisher English training speech parts 1 and 2. Philadelphia: Linguistic Data Consortium, 2004.
[14] 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(4):357–366, 1990.
[15] N. Dehak, P. Dumouchel, and P. Kenny. Modeling prosodic features with joint factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing, 15(7):2095–2103, 2007.
[16] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing, 19(4):788–798, 2011.
[17] G. Doddington. Speaker recognition based on idiolectal differences between speakers. In Seventh European Conference on Speech Communication and Technology, 2001.
[18] C. Espy-Wilson, S. Manocha, and S. Vishnubhota. A new set of features for text-independent speaker identification. In Ninth International Conference on Spoken Language Processing, 2006.
[19] 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), pages 347–352. IEEE, 2015.
[20] 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.
[21] 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.
[22] D. Fleming, B. L. Giordano, R. Caldara, and P. Belin. A language-familiarity effect for speaker discrimination without comprehension. Proceedings of the National Academy of Sciences, 111, 2014.
[23] L. F. Gallardo, M. Wagner, and S. Möller. Spectral sub-band analysis of speaker verification employing narrowband and wideband speech. Citeseer.
[24] D. García-Romero and C. Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In INTERSPEECH, volume 2011, pages 249–252, 2011.
[25] J. Gudnason and M. Brookes. Voice source cepstrum coefficients for speaker identification. In ICASSP, pages 4821–4824. IEEE, 2008.
[26] J. Guo, R. Yang, H. Ariskere, 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.
[27] H. Hermansky. Perceptual linear predictive (plp) analysis of speech. The Journal of the Acoustical Society of America, 87(4):1738–1752, 1990.
[28] X. Huang, A. Acero, and H. Hon. Spoken language processing: A guide to theory, algorithm, and system development, volume 95.
[29] X. Kawakami, L. Wang, A. Kai, and S. Nakagawa. Speaker identification by combining various vocal tract and vocal source features. In Text, Speech and Dialogue. Springer, 2014.
[30] C. Kim and R. Stern. Power-normalized cepstral coefficients (PNCC) for robust speech recognition. In ICASSP, pages 4101–4104. IEEE, 2012.
[31] T. Kinnunen. Long-term F0 modeling for text-independent speaker recognition.
[32] T. Kinnunen. Spectral features for automatic text-independent speaker recognition. Licentiate’s thesis, 2003.
[33] T. Kinnunen and H. Li. An overview of text-independent speaker recognition: From features to supervisors. Speech Communication, 52(1):12–40, 2010.
[34] G. Klabmayer, T. Unterrhiner, A. Mayr, and S. Hochreiter. Self-normalizing neural networks. arXiv preprint arXiv:1706.02515, 2017.
[35] N. Krishnamurthy and J. Hansen. Babble noise: modeling, analysis,
and applications. IEEE Transactions on Audio, Speech, and Language Processing, 17, 2009.

[36] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu. Deep speaker: an end-to-end neural speaker embedding system. arXiv preprint arXiv:1705.02364, 2017.

[37] L. Li, D. Wang, A. Rozzi, 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.

[38] L. Lu, Y. Dong, X. Zhao, J. Liu, and H. Wang. The effect of language factors for robust speaker recognition. In ICASSP. IEEE, 2009.

[39] R. Mammon, X. Zhang, and R. Ramachandran. Robust speaker recognition: A feature-based approach. Signal Processing Magazine, 13, 1996.

[40] B. Milner and X. Shao. Speech reconstruction from mel-frequency cepstral coefficients using a source-filter model. In INTERSPEECH, pages 2421–2424, 2002.

[41] V. Mitra, H. Franco, M. Graciarena, and A. Mandal. Normalized amplitude modulation features for large vocabulary noise-robust speech recognition. In ICASSP, pages 4117–4120. IEEE, 2012.

[42] P. Mowlane, R. Saedi, and Y. Stylianou. Phase importance in speech processing applications. In ISCA, 2014.

[43] H. Muckenhirn, V. Abrol, M. M. Doss, and S. Marcel. Understanding and visualizing raw waveform-based CNNs. In INTERSPEECH, 2019.

[44] H. Muckenhirn, M. M. Doss, and S. Marcel. Towards directly modeling raw speech signal for speaker verification using CNNs. In ICASSP, pages 4884–4888. IEEE, 2018.

[45] H. Muckenhirn, M. Magimai-Doss, and S. Marcel. End-to-end convolutional neural network-based voice presentation attack detection. In IJCB, pages 335–341. IEEE, 2017.

[46] L. Muda, M. Begam, and I. Elamvazuthi. Voice recognition algorithms with mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. CoRR, abs/1003.4083, 2010.

[47] K. Murty and B. Yegnanarayana. Combining evidence from residual L. Muda, M. Begam, and I. Elamvazuthi. Voice recognition algorithms with mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. CoRR, abs/1003.4083, 2010.

[48] K. Murty and B. Yegnanarayana. Combining evidence from residual phase and MFCC features for speaker recognition. Signal Processing Letters, 13(1):52–55, 2006.

[49] A. v.d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative flow for raw audio. arXiv preprint arXiv:1609.03499, 2016.

[50] O. D. Orman and L. M. Arslan. Frequency analysis of speaker identification. In A Speaker Odyssey-The Speaker Recognition Workshop, 2001.

[51] H. Parthasarathi, R. Padmanabhan, and H. Murthy. Robustness of group delay representations for noisy speech signals. International Journal of Speech Technology, 14(4):361, 2011.

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

[53] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al. The kaldi speech recognition toolkit. 2011.

[54] M. Ravanelli and Y. Bengio. Speaker recognition from raw waveform with sincnet. In Spoken Language Technology Workshop (SLT), pages 1021–1028. IEEE, 2018.

[55] D. Reynolds. Experimental evaluation of features for robust speaker identification. IEEE Transactions on Speech and Audio Processing, 2(4):639–643, 1994.

[56] A. Ross, S. Banerjee, C. Chen, A. Chowdhury, V. Mirjalili, R. Sharma, T. Swearingen, and S. Yadav. Some research problems in biometrics: The future beckons. In A Speaker Odyssey-The Speaker Recognition Workshop, 2001.

[57] A. Ross, S. Banerjee, C. Chen, A. Chowdhury, V. Mirjalili, R. Sharma, T. Swearingen, and S. Yadav. Some research problems in biometrics: The future beckons. In A Speaker Odyssey-The Speaker Recognition Workshop, 2001.

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

[59] 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.

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

[61] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In CVPR. IEEE, 2015.

[62] R. Singh. Profiling humans from their voice. Springer, 2019.

[63] T. Virtanen, R. Singh, and B. Raj. Techniques for noise robustness in automatic speech recognition. John Wiley & Sons, 2012.

[64] E. Wong and S. Sridharan. Comparison of linear prediction cepstrum coefficients and mel-frequency cepstrum coefficients for language identification. In Intelligent Multimedia, Video and Speech Processing, Proceedings of International Symposium on, pages 95–98. IEEE, 2001.

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

[66] Y. Zhou and Z. Zhao. Fast ICA for multi-speaker recognition system. In International Conference on Intelligent Computing, pages 507–513. Springer, 2010.