DeepVOX: Discovering Features from Raw Audio for Speaker Recognition in Non-ideal Audio Signals

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

Abstract—Automatic speaker recognition algorithms typically use pre-defined filterbanks, such as Mel-Frequency and Gammatone filterbanks, for characterizing speech audio. However, it has been observed that the features extracted using these filterbanks are not resilient to diverse audio degradations. In this work, we propose a deep learning-based technique to deduce the filterbank design from vast amounts of speech audio. The purpose of such a filterbank is to extract features robust to non-ideal audio conditions, such as degraded, short duration, and multi-lingual speech. To this effect, a 1D convolutional neural network is designed to learn a time-domain filterbank called DeepVOX directly from raw speech audio. Secondly, an adaptive triplet mining technique is developed to efficiently mine the data samples best suited to train the filterbank. Thirdly, a detailed ablation study of the DeepVOX filterbanks reveals the presence of both vocal source and vocal tract characteristics in the extracted features. Experimental results on VOXCeleb2, NIST SRE 2008, 2010 and 2018, and Fisher speech datasets demonstrate the efficacy of the DeepVOX features across a variety of degraded, short duration, and multi-lingual speech. The DeepVOX features also shown to 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

Automatic 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 [5] and telephone banking systems [10]. VUIs are gaining popularity due to the ease-of-access provided by their hands-free operation. However, in practice, the voice input to the speaker recognition systems often exhibits non-ideal speech audio characteristics, such as degraded [14], multi-lingual [36], and short-duration [27] speech. The unfavorable nature of these non-ideal inputs propagates through different components of the speaker recognition system and lowers its performance [14], [46]. Therefore, it is important to develop speaker recognition techniques that are robust to a wide variety of non-ideal audio conditions, thereby providing generalizable speaker recognition performance.

Some of the current speaker recognition enabled consumer devices address the issues of non-ideal audio conditions at the sensor-level by employing specialized hardware, such as far-field microphone arrays [5]. However, the use of specialized hardware interfaces limits their backward compatibility with existing speaker recognition systems. On the other hand, some of the latest speaker recognition techniques address these issues at the software-level by designing robust matchers [13], [14], [51]. However, these techniques rely on traditional handcrafted speech features such as Mel-Frequency Cepstral Coefficients (MFCC) and Linear Predictive Coding (LPC), whose representation capability varies with the quality of input audio, thus limiting the effectiveness of the subsequent matcher [25]. While some of the recent work in end-to-end speaker recognition can perform speaker recognition directly from raw input audio, their robustness to non-ideal audio conditions is yet to be determined [25], [44].

We position our work with the existing literature by approaching the issue of non-ideal audio conditions at the feature-level. We design a raw audio-based speech feature extractor, called DeepVOX, that is robust to non-ideal audio conditions and 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. The main contributions in this work are as follows:

1) We propose a Convolutional Neural Network (CNN) based approach for learning a robust speech filterbank, referred to as DeepVOX, directly from raw speech audio.
2) We propose an adaptive triplet mining technique for efficiently training the DeepVOX filterbank in conjunction with 1D-Triplet-CNN [14], 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 existing speaker recognition algorithms such as the xVector-PLDA [51] and the iVector-PLDA [17].
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 to identify the type of speech features extracted by DeepVOX and characterize their frequency-response to a various degraded speech audio.

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 and was later observed to be efficient for performing speaker recognition as well [55]. However, MFCC features are not robust to audio degradations [23] and are, therefore, not very suitable for speaker recognition tasks in presence of noisy speech data. This lack of generalizability in the performance of the handcrafted features, such as MFCC, primarily stems from the fact that they are derived from auditory experiments of limited scale. Although these auditory experiments have been later revised multiple times [42], [43] to improve their robustness to a wider variety of audio conditions, the scope for improvement remains. This motivated the development of robust speech features for performing speaker recognition in varied non-ideal audio conditions. These speech features are also adept at encoding various physical and acoustic properties of human voice and can be accordingly partitioned into several feature categories, as follows [31].

- Short-term spectral features encode the vocal tract shape.
- Vocal source features characterize the glottal excitation signal.
- Prosodic features model the speaking style of a speaker.
- High-Level Features model the lexicon of a speaker.

According to the source-filter model of speech [37], 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 for extracting stable voice characteristics are based on either MFCC or Linear Predictive Coding (LPC) [29]. The MFCC feature extraction process uses triangular-filters placed on the Mel-scale for modeling the human auditory perception system [40]. The LPC, on the other hand, estimates an all-pole model of filter design for modeling the vocal tract [37].

Humans are noted to be efficient in performing speaker recognition in the presence of unknown audio degradations. However, the MFCC feature, which is based on human auditory processing, is unable to cope well in such scenarios [58]. Motivated by this, the authors in [57] propose the Gammatone Filterbank-based Gammatone Frequency Cepstral Coefficients (GFCC) features as a noise-robust alternative to MFCC. 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 potentially improve speaker recognition performance. 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 this was based on human auditory system experiments [19], where short-term phase spectrum did not provide enough performance benefits to justify the associated computational expenses. However, recent studies have reported comparable and complementary speaker recognition performance of both magnitude- and phase-based features [41], [47].

LPC-based methods [56] 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 [56]. 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. [24] 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 [24] 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 both 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 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, and fundamental frequency can potentially be extracted [18] to perform speaker recognition. One such work in [59] used an LPC-based inverse vocal tract filter and wavelet transform for extracting vocal source features called Wavelet Octave Coefficients Of Residues (WOCOR). It also combined MFCC and WOCOR features for improving overall speaker verification performance.

In the past decade, deep learning based methods have been successfully designed and implemented for solving many speech processing tasks, including speaker recognition [13], [14], [34]. A majority of such speaker recognition methods use some type of hand-crafted features, e.g. MFCC, LPCC, as input to their network for solving the problem. For example, authors in [34] 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 [39], [44] have proposed to feed the raw speech waveform directly as input to deep neural networks for performing a variety of tasks such as speaker recognition and detection of voice presentation attacks. The authors in [44], 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 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 [13] and 1D-Triplet-CNN [14] 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 robust and highly discriminative speech audio features.

Note that, unlike the work in [44], 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 [26], [39], [44], we demonstrate the compatibility of the proposed DeepVOX features with state-of-the-art deep learning-based speaker recognition methods such as xVectors [51] and 1D-Triplet-CNN [14] and even on classical methods such as the iVector-PLDA [17].

3 PROPOSED ALGORITHM

In the previous section, we discussed some popular speech feature extraction techniques. Depending upon the type of the features being extracted, the algorithms were further categorized into four different feature categories. 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 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 Speech Feature Extraction Using DeepVOX

In this work, we use the proposed DeepVOX feature extractor jointly with a 1D-Triplet-CNN [14]-based feature embedding network for performing speaker recognition. The 1D-Triplet-CNN [14] 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

We first use a Voice Activity Detector [7] to remove non-speech parts of an input audio. Any data sample longer than 2 seconds is split into multiple smaller 2 second long audio samples. The resulting speech audio is then framed and windowed into multiple smaller audio clips, called speech units, using a hamming window of length 20ms and stride 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 2 second long audio sample. These speech units are then stacked horizontally to form a two-dimensional speech audio representation called speech frame, of dimension 160 × 200. The extracted speech frames are then made into speech frame triplets and fed into DeepVOX.

3.1.2 Speech Frame Triplets

The authors in [50] introduced the idea of triplet based CNNs. As illustrated in Figure 2, our DeepVOX architecture takes a speech frame triplet Di as input. A speech frame triplet Di is defined as a tuple of three speech frames: Di = (Sa, Sp, Sn) Here, Sa, the anchor sample, and Sp, the positive sample, are two different speech samples from a subject ‘X’, Sn, the negative sample, is a speech sample from another subject ‘Y’, such that X ≠ Y.

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 [14] and SELU [33] (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 interfaced with SELU non-linearity here are
The 1D-Triplet-CNN architecture 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 [42], where the authors use dilated convolutions to increase the receptive field size nonlinearly with a linear increase in number of parameters. In context of 1D-Triplet-CNN, 1D dilated convolutions allow the network to learn sparse relationships between the feature values within a speech unit leading to significant performance benefits. The 1D-Triplet-CNN architecture [14] is designed for learning speaker dependent characteristics. The three parallel network branches in the 1D-Triplet-CNN architecture learn and share a common set of weights (see Figure 3). 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.

3.1.4 1D-Triplet-CNN

The cosine triplet embedding loss is a modification upon the triplet loss initially introduced in [50] by replacing the euclidean distance metric with cosine similarity. The triplet loss is designed to learn an embedding \( g(f(x)) \in \mathbb{R}^d \), where \( f(x) \) is DeepVOX feature of speech frame \( x \). In this work, \( d \) is set to 128. The embedding is so learned that the intra-class samples are embedded closer to each other than the inter-class samples. The mathematical formulation of cosine triplet embedding loss is given by:

\[
L(S_a, S_p, S_n) = \sum_{a,p,n} \left( \cos(g(f(S_a)), g(f(S_p))) - \cos(g(f(S_a)), g(f(S_n))) + \alpha_m \right)
\]  

Here, \( L(\cdot, \cdot, \cdot) \) is the cosine triplet embedding loss function. \( S_a \) (the anchor sample) and \( S_p \) (the positive sample) are two different speech samples from a subject ‘X’. \( S_n \) (the negative sample) is a speech sample from another subject ‘Y’, such that \( X \neq Y \). \( N \) refers to the total number of triplets in the training set. \( \alpha_m \) is the margin of the minimum distance between positive and negative samples and is a user tunable hyper-parameter.

In the training phase, the triplet loss helps the network learn the similarity between the anchor and the positive samples and dissimilarity between the anchor and the negative samples. As illustrated in Figure 2, we train both the DeepVOX and the 1D-Triplet-CNN networks together thus simultaneously learning the embedding space using the 1D-Triplet-CNN and the feature space using the DeepVOX.

In the testing phase (see Figure 2) 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. During testing, we input a pair of speech samples into the siamese network to extract a corresponding pair of speech embeddings. The speech embedding pair is then compared using the cosine similarity metric to render a match score. Under ideal conditions, the match score for a genuine pair should be close to 1, while the match score for an impostor pair should be close to \(-1\).

3.1.5 Cosine Triplet Embedding Loss

The effectiveness and generalizability of any network trained using the triplet learning paradigm, such as 1D-Triplet-CNN [14], depends on the difficulty of the training triplets. The authors in [14] 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 [50], we propose a curriculum learning-based [11] adaptive triplet mining technique.
In adaptive triplet mining, at a given epoch $i$, the goal is to select a negative sample $S_n^i$, such that:

$$\cos(g(f(S_n^i)), g(f(S_p^i))) > \cos(g(f(S_n^i)), g(f(S_n^i))) + \alpha_m.$$  \hspace{1cm} (2)

Here, $S_n^i$ is the anchor speech sample, $S_p^i$ is the positive speech sample, and $\alpha_m$ is the margin. Here, $\tau_{S_n}$ is a parameter that denotes the average difficulty of $S_n^i$ (a negative sample), chosen at epoch $i$. The difficulty of a negative sample is computed using its cosine similarity to the corresponding anchor speech sample in the triplet. Harder negative samples typically have higher cosine similarity to the corresponding anchor samples, making them harder to separate from the anchor samples. A value of $\tau = 0$ yields the easiest negative sample and $\tau = 1$ yields the hardest negative sample. In our experiments, the value of $\tau$ is determined by the current stage (or epoch) of the training process. We initialize the training with the value of $\tau$ at 0.4 (empirically chosen) and increase it gradually to 1.0 through the course of the training. This is done to ensure a minimum difficulty of the training triplets at the beginning of the training which is gradually increased as the training proceeds. This helps in avoiding the problem of bad local minima caused by introducing harder negative triplets directly at the beginning of the training \cite{50}. It is also observed that learning only on easy and semi-hard triplets lead to poor generalization capability of the model on harder evaluation pairs. Additionally, the model is pre-trained in the identification mode to ensure easier initialization of the training process.

### 3.2 Analysis of the DeepVOX Architecture

In Section 3.1.3, we introduced the DeepVOX architecture for extracting short-term spectral features. In this section, we mathematically analyze the proposed architecture and compare DeepVOX’s feature learning process with some popular short-term spectral feature extraction algorithms such as MFCC, PNCC, PLP and MHEC. However, before proceeding with the mathematical analysis of DeepVOX’s network architecture, we first draw a visual comparison with some popular short-term spectral feature extraction algorithms in Figure 3. 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. We further use this comparative study to explain the similarities and dissimilarities between our proposed algorithm and some of the existing short-term spectral feature extraction algorithms.

Furthermore, please note that the DeepVOX method is proposed as an alternative for short-term spectral features such as MFCC and LPC and is intended to be used alongside feature embedding methods such as xVector, iVector, or 1D-Triplet-CNN for performing speaker recognition. Therefore, DeepVOX, similar to MFCC and LPC, is strictly a short-term time-domain feature extraction method, whereas xVector, iVector, and 1D-Triplet-CNN are speech feature embedding methods. Additionally, DeepVOX features, unlike the xVector embeddings, are not a mid-level representation drawn from an end-to-end speaker recognition neural network. Instead, DeepVOX is an independent neural network model carefully designed to learn a time-domain speech filterbank directly from raw audio data. Such an approach makes the DeepVOX features, unlike existing deep learning-based speech embedding networks \cite{29, 51}, a direct alternative for short-term spectral features such as MFCC and LPC in speaker recognition models. We specifically trained xVector and iVector models using DeepVOX features to demonstrate its compatibility with existing deep learning-based and classical speaker recognition methods. The experimental results given Section 5 show its performance benefits over MFCC, LPC, and MFCC-LPC features.

#### 3.2.1 Building Blocks of Short-term Spectral Feature Extraction Algorithms

The comparison in Figure 3 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 high-pass filtered to compensate for the natural suppression of high frequency components in the human voicebox. However, this step can degrade the speech quality if the input audio has high-frequency noise and is therefore skipped in our proposed method.
- **Framing and Windowing:** Next, 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. In our case, we use a hamming window of length of 20ms and a stride of 10ms for framing and windowing the speech signal.
- **Fourier Transform:** FFT (Fast Fourier Transform) is performed to decompose a speech signal based on its frequency content. However, in our case, instead of decomposing the speech frames into their frequency components using FFT, DeepVOX learns speech features in the time domain itself.
- **Filterbank Integration:** The FFT response is usually processed through various handcrafted filterbanks (such as Mel filterbank) for extracting the individual speech features. However, for DeepVOX, instead of using handcrafted filterbanks, a non-linear combination of multiple convolutional filters is used to learn filterbanks specifically suited for performing speaker recognition.
- **Non-linear Rectification:** This step is done to compress the dynamic range of filterbank energies to improve speaker recognition performance \cite{55}. 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 and PNCC extract the speaker dependent features from a speech signal using handcrafted filterbanks. To this effect, the Fourier Transform is used to decompose a speech signal into its constituent frequencies, thereby, making filtering operation easier, both semantically and computationally. As the computationally-expensive convolution operation, between the signal and the filter, in time domain is replaced by pointwise multiplication in the frequency domain. The Fourier Transform is usually implemented using the Fast Fourier Transform (FFT) algorithm which makes the filtering of 1D audio signals even more computationally efficient, $O(n \log n)$, as compared to general convolution operation, $O(n^2)$. However, FFT only provides a close approximation of time domain filtering and is often inconsistent across different implementations \cite{19}, thereby enforcing a trade-off between computational complexity and accuracy. Furthermore, the recent development of efficient GPU-driven implementations of the convolution operation makes Convolutional Neural Networks (CNN) extremely well-suited for performing time domain filtering. Therefore, we use CNN in our algorithm to learn time-domain filters efficiently from raw speech audio.
As discussed earlier and illustrated in Figures 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:

$$S = \{u_1, u_2, \ldots, u_i, \ldots, u_n\}. \quad (4)$$

Here $u_i$ is the $i^{th}$ speech unit and $n$ is the total number of speech units, in the speech frame $S$. Furthermore, the network outputs a 40 channel filter response $f_i$ corresponding to every speech unit $u_i$. Therefore, DeepVOX’s output $O$ can be given by:

$$O = \{f_1, f_2, \ldots, f_i, \ldots, f_n\}. \quad (5)$$

$$f_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,j}, \ldots, x_{i,40}\}^T \quad (6)$$

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 in the network. Therefore, the network output $f_i$ corresponding to speech unit $u_i$ can be written as:

$$f_i = l_m \cdot (l_{m-1} \cdot \cdots \cdot l_k \cdot l_1(u_i)). \quad (7)$$

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:

$$l_k(u_i) = C_k \odot u_i, \quad (8)$$

Here $C_k$ is the convolutional filter for the $k^{th}$ layer. The operation in Eq. (8) is equivalent to time-domain filtering of input signal $u_i$ with filter $C_k$. Hence, we can rewrite Eq. (7) as:

$$f_i = (C_m \odot (C_{m-1} \odot \cdots \cdot C_k \odot \cdots \cdot C_1 \odot u_i)). \quad (9)$$

Since the convolution operation is associative, we can rewrite Eq. (9) as:

$$f_i = (C_m \odot C_{m-1} \odot \cdots \cdot C_k \odot \cdots \cdot C_1) \odot u_i; \quad (10)$$

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.

4 Datasets and Experiments

In this work, we perform multiple speaker verification experiments on a variety of datasets and protocols. Primarily, we use the VOXCeleb2 [15], Fisher English Training Speech Part 1 [16], and NIST SRE (2008 [1], 2010 [2], and 2018 [3]) datasets for training and evaluating the proposed and baseline speaker verification algorithms. We also create degraded versions of the Fisher and NIST SRE 2008 speech datasets by adding diverse noise data from the NOISEX-92 [55] 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 diverse audio degradations. Additionally, all the speech datasets were sampled at a rate of 8kHz to match the NIST SRE dataset specifications [1].

4.1 Datasets

4.1.1 VOXCeleb2 Dataset

The VOXCeleb2 [15] dataset contains short interview video clips of 6, 112 celebrities recorded in unconstrained scenarios. The entire VOXCeleb2 dataset contains 145, 569 video samples from 5, 994 celebrities in the training set and 4, 911 videos from the remaining 118 speakers in the evaluation set. However, for keeping the triplet-based training process computationally tractable, we only use speech data from one randomly selected video for each subject. In our experiments, each video in the dataset is processed to extract the speech audio, sampled at 8000Hz, from its audio track. Speech samples longer than 5 seconds are split into multiple non-overlapping 5 second long speech samples.

4.1.2 Fisher English Training Speech Part 1 Dataset

The Fisher dataset 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 [55] noise dataset to the Fisher speech dataset. The resultant ‘degraded-Fisher’ speech dataset was maintained at a SNR level of 10dB. We also added reverberations to the speech data generated in a simulated cubical room of side length 4m.

4.1.3 NIST SRE 2008, 2010, and 2018 Datasets

We also use the NIST SRE 2008 [1] dataset in our experiments, given in Table 3 and Figure 4 to evaluate the performance of our proposed algorithm in the presence of multi-lingual data. 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. As mentioned earlier, we also add F-16 and Babble noise at a resultant SNR of 0dB to the NIST SRE 2008 dataset to vastly increase the difficulty of the task. We also perform cross-dataset speaker verification performance evaluation using speech data from all the speakers in the evaluation sets of the NIST SRE 2010 [2] and NIST SRE 2018 [3] datasets.

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 [1] [2] and [3]. The noise characteristics of the training and testing sets used in the different experiments are given alongside in Tables [1] [2] and [3]. For example, in Experiment 3 given in Table [3] the model was trained on speech data from the training set of the Fisher Speech Dataset degraded with Babble noise, and the evaluation was done on speech data from testing set of Fisher Speech Dataset degraded with F16 noise. Note that, no mention of a noise type, such as in Experiment 1 given in Table [1] indicates usage of un-altered speech data from the original dataset. Additionally, we have also conducted speaker verification experiments on a subset of multi-lingual speakers from the NIST SRE 2008 dataset, as shown in Table [4] for evaluating the effect of speech language on speaker verification performance. Finally, as illustrated in Figure [7] and discussed in Section [9] we have performed Guided Backpropagation [2] based ablation study of the features extracted by trained DeepVOX models, to understand the type of audio features considered important for performing speaker recognition by the DeepVOX model.

### 4.2.1 Baseline Speaker Verification Experiments

For establishing baseline speaker verification performance on the VOXCeleb2, Fisher, NIST SRE 2008, 2010, and 2018 speech datasets, we choose iVector-PLDA [22] and xVector-PLDA [51] algorithms trained on the baseline features (MFCC, LPC, MFCC-LPC) and DeepVOX features separately. This is done to evaluate and compare the effectiveness of DeepVOX features, with respect to baseline features, in both classical and deep learning-based speaker recognition algorithms. However, unlike the baseline features, DeepVOX feature extraction process requires a DeepVOX model to be trained. For each of the experiments in Tables [1] [2] and [3] we use speech data only from corresponding training set to train the DeepVOX model, ensuring disjoint data and subjects in the training and testing sets for the DeepVOX feature extraction process. We also use the RawNet2 [26] algorithm for establishing baseline raw audio-based speaker recognition performance.

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

We also perform speaker recognition experiments using the 1D-Triplet-CNN [14] algorithm. These experiments provide benchmark results (given in Tables [12] and [3]) to directly compare the
The proposed DeepVOX features outperform the baseline features for a majority of methods and data partitions, across all the metrics.

### Table 3

| # | Train set/Test set | Method | TMR@FMR=1 (%, %) | minDCF (target (0.001, 0.01)) | Error Equal (EER, in %) |
|---|---|---|---|---|---|
| 6 | P1 / P1 | M1 | 55.21, 93.06 | 44.09, 87.52 | 52.50, 93.22 | 81.65, 97.63 | 0.097, 0.76 | 0.064, 0.84 | 0.095, 0.89 | 0.067, 0.89 | 8.74 | 11.18 | 8.15 |
| 7 | P1 / P2 | M1 | 53.17, 92.72 | 49.11, 86.65 | 60.21, 93.36 | 81.37, 97.30 | 0.098, 0.72 | 0.085, 0.63 | 0.075, 0.76 | 0.066, 0.59 | 10.55 | 14.62 | 8.34 |
| 8 | P1 / P3 | M1 | 25.30, 76.40 | 22.36, 76.47 | 24.05, 85.24 | 23.97, 87.78 | 0.099, 0.79 | 0.068, 0.83 | 0.100, 0.98 | 0.099, 0.92 | 14.15 | 15.13 | 11.95 |
| 9 | P4 / P4 | M1 | 46.70, 81.51 | 38.04, 87.86 | 42.16, 88.38 | 37.63, 96.12 | 0.097, 0.67 | 0.067, 0.87 | 0.095, 0.93 | 0.094, 0.93 | 12.37 | 13.85 | 10.81 |
| 10 | P5 / P5 | M1 | 8.40, 24.93 | 7.58, 23.56 | 8.40, 24.47 | 8.44, 24.10 | 0.098, 0.66 | 0.097, 0.98 | 0.096, 0.96 | 0.069, 0.98 | 43.29 | 41.65 | 43.74 |
| 11 | P4 / P5 | M1 | 2.26, 14.61 | 1.85, 14.54 | 2.43, 25.20 | 2.57, 24.13 | 0.097, 0.97 | 0.097, 0.97 | 0.099, 0.99 | 0.099, 0.98 | 45.02 | 44.11 | 39.40 |
| 12 | P5 / P5 | M1 | 12.67 | 12.28 | 12.67 | 12.28 | 0.098, 0.97 | 0.097, 0.99 | 0.097, 0.99 | 0.097, 0.99 | 43.84 | 46.79 | 45.52 |
| | | M2 | 3.19, 16.35 | 3.74, 17.26 | 3.19, 16.10 | 1.19, 10.14 | 3.77, 15.54 | 0.098, 0.98 | 0.099, 0.99 | 0.099, 0.99 | 0.099, 0.99 | 44.75 | 44.29 | 47.40 |
| | | M3 | 0.099, 0.99 | 0.099, 0.99 | 0.099, 0.99 | 0.099, 0.99 | 0.100, 1.00 | 0.100, 1.00 | 0.100, 1.00 | 0.100, 1.00 | 0.100, 1.00 | 0.100, 1.00 |
| | | M4 | 5.35, 22.57 | 5.32, 22.44 | 5.41, 22.04 | 4.18, 26.64 | 5.72, 24.57 | 0.099, 0.99 | 0.099, 0.99 | 0.100, 1.00 | 0.100, 1.00 | 0.100, 1.00 | 0.100, 1.00 |
| | | M5 | 35.28, 83.49 | 38.01, 81.99 | 35.25, 86.66 | 78.16, 94.46 | 0.098, 0.88 | 0.098, 0.99 | 0.098, 0.99 | 0.098, 0.99 | 12.67 | 14.44 | 11.69 |
| | | M6 | 39.38, 84.36 | 35.48, 80.49 | 37.92, 90.00 | 67.22, 95.58 | 0.099, 0.90 | 0.099, 0.97 | 0.102, 0.94 | 0.102, 0.94 | 12.94 | 14.24 | 10.70 |
| | | M7 | 22.44, 75.09 | 20.81, 65.42 | 23.64, 72.66 | 24.17, 63.72 | 0.099, 0.99 | 0.095, 0.95 | 0.100, 1.00 | 0.100, 1.00 | 15.24 | 19.24 | 16.17 |
| | | M8 | 19.57, 72.87 | 19.58, 72.46 | 19.70, 81.25 | 31.80, 83.67 | 0.099, 0.99 | 0.093, 0.83 | 0.106, 0.99 | 0.106, 0.99 | 13.53 | 17.34 | 28.14 |

**Performance of DeepVOX features compared to MFCC, LPC, and MFCC-LPC features in a deep learning framework. For training the 1D-Triplet-CNN, speech audio triplets are formed using the speakers from the training set. The speech audio triplets are then processed to extract 40 dimensional MFCC and LPC features separately. The extracted MFCC and LPC features are then stacked together to form a two-channel input feature patch for the 1D-Triplet-CNN. For evaluation, speech audio pairs are fed to the trained model to generate pairs of speech embeddings. The speech embeddings are then matched using the cosine similarity metric.**

### 4.2.3 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 12 and 3) drawn from different datasets and noise types and compare it with the baseline algorithms. Similar to the MFCC-LPC-based 1D-Triplet-CNN algorithm, our algorithm also trains on speech audio triplets. However, instead of extracting hand-crafted features like MFCC or LPC, our algorithm trains the DeepVOX and 1D-Triplet-CNN modules together to learn both the DeepVOX-based feature representation and 1D-Triplet-CNN-based speech feature embedding simultaneously. For evaluation, speech audio pairs are fed to the trained DeepVOX model to extract pairs of DeepVOX features which are then fed into the trained 1D-Triplet-CNN model to extract pairs of speech embeddings and compare them using the cosine similarity metric.

**4.2.4 1D-Triplet-CNN-based Speaker Recognition Experiments Using Adaptive Triplet Mining**

The proposed adaptive triplet mining technique is evaluated by repeating all the 1D-Triplet-CNN based speaker verification experiments on MFCC, LPC, MFCC-LPC, and DeepVOX features, referred to as 1D-Triplet-CNN online in Tables 12 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 (τ) of the mined negative samples is gradually increased from 0.4 to 1.0 linearly over 800 epochs. Also, it is important to note that the triplet mining is done in mini-batches of 6 randomly chosen samples drawn from each of the 25 randomly chosen training subjects.

**4.2.5 Effect of Language on Speaker Verification**

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 [36]. 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 [36]. In this work, we perform additional speaker recognition experiments (Exp. # 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
Verification Results on multi-lingual speakers from the NIST SRE 2008 dataset. The proposed DeepVOX features outperform the baseline features for a majority of methods and data partitions, across all the metrics.

| # | Train set / Test set | Method | TMR@FMR=1% | minDCF (p=0.01) | Equal Error Rate (EER, in %) |
|---|---------------------|--------|-------------|-----------------|-----------------------------|
| 13 | L1/L3 | MFCC | LPC | MFCC- L1 | LCC- L1 | MFCC- L1 | LCC- L1 | MFCC- L1 | LCC- L1 | MFCC- L1 | LCC- L1 |
| M4 | 15.24, 10.56 | 72.26, 97.87 | 0.096, 0.86 | 0.075, 0.63 | 15.26, 7.81 | 71.26, 94.96 | 0.094, 0.86 |
| M5 | 16.56, 11.89 | 72.26, 97.87 | 0.096, 0.86 | 0.075, 0.63 | 15.26, 7.81 | 71.26, 94.96 | 0.094, 0.86 |

**Same language, english only trials**: In Exp. # 12, the trained models are evaluated on same-language (English Only) trials. This experiment establishes the baseline same-language (English to English) speaker verification performance of all the algorithms.

**Same language, non-english trials**: In Exp. # 13, the trained models are evaluated on same-language (Multi-lingual) trials. 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. Hindi to Hindi) speech trials.

**Cross-lingual trials**: In Exp. # 14, the trained models are evaluated on different-language (Cross-lingual) trials. 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 Effect of Audio Length on Speaker Verification**

The reliability of the speaker-dependent features extracted from an audio sample depends on the usable speech data present within, which is directly dependent on the length of the audio sample. Therefore, performing speaker recognition in audio samples of a small duration is a challenging task. Since in real-life scenarios, probe audios are of relatively small audio durations (1 sec - 3 secs), the feature extraction algorithm needs to be able to reliably extract speaker-dependent features from speech audio of limited duration. In this experiment (see Table 4 and Figure 4), we compare the speaker verification performance of our proposed algorithm with the baseline algorithms on speech data of varying duration from the NIST SRE 2008 dataset. The duration of probe audio is varied between 3.5 secs and 0.5 secs in steps of 0.5 secs.

**5 Results and Analysis**

The results for all the experiments described in Section 4.2 are given in Tables 4 and 5. The content of all the tables is summarised in the DET curves given in Figures 4 and 5 to present the results in an easier-to-consume format. For all the speaker verification experiments, we report the Match Rate at False Match Rate of 1% and 10% (TMR@FMR={1%, 10%}), minimum Detection Cost Function (minDCF) at $C_{miss}$ (cost of a missed detection) value of 1 and Equal Error Rate (EER, in %) as our performance metrics for comparison of the baseline methods and the proposed method. The minDCF is reported at two different a priori probability of the specified target speaker, viz., $P_{tar} = 0.01$ and 0.001 (minDCF($P_{tar} = \{0.01, 0.001\}$)). The Detection Error Tradeoff (DET) curves are given in Figure 4.

- Overall, in all the speaker verification experiments given in Tables 4 and 5, the 1D-Triplet-CNN algorithm using DeepVOX features trained with adaptive triplet mining, also referred to as 1D-Triplet-CNN online (DeepVOX), performs the best. The proposed adaptive triplet mining method improves the verification performance (TMR@FMR=1%) of the 1D-Triplet-CNN algorithm using DeepVOX features by 3.01%, and MFCC-LPC features by 8.71%. Similar performance improvements are also noticed for the MFCC and LPC features across all the performance metrics. This establishes the benefits of using the adaptive triplet mining technique over offline-triplet mining for efficiently training the 1D-Triplet-CNN based speaker recognition models.

- Across all the speaker verification experiments given in Tables 4 and 5, the second-best performance, after DeepVOX features, is obtained by the feature level combination of MFCC and LPC features, referred to as MFCC-LPC features. Therefore, we choose MFCC-LPC features as our strongest baseline feature. In the upcoming discussions, all performance improvements offered by the DeepVOX features, for any particular algorithm, is reported in comparison to the MFCC-LPC features. Furthermore, we will also draw comparison with the RawNet2 model to establish DeepVOX's performance benefits over a current state-of-the-art raw speech audio-based speaker recognition method.

- In experiment #1 on the VOXCeleb2 dataset, given in Table 1 and Figure 4, the 1D-Triplet-CNN online (DeepVOX) method performs the best across all the performance metrics. The DeepVOX features improve the verification performance (TMR@FMR={1%, 10%}), specifically for the 1D-Triplet-CNN online algorithm, over the best performing baseline feature (MFCC-LPC) by $\{0.89, 0.9\%\}$. It also reduces the EER by 2.5% and minDCF($P_{tar} = \{0.001, 0.01\}$) by $\{0.03, 0.15\}$. Similarly, for the 1D-Triplet-CNN, xVector-PLDA and iVector-PLDA algorithm, the DeepVOX features improve verification performance over the best performing baseline feature (MFCC-LPC). The 1D-Triplet-CNN (DeepVOX) method also outperforms the RawNet2 across all the performance metrics. The
Table 5
Verification Results under varying audio length on the NIST SRE 2008 dataset. The proposed DeepVOX features outperform the baseline features for a majority of methods and data partitions, across all the metrics.

| Length (secs) | Method | M1 | M2 | M3 | M4 | M5 |
|---------------|--------|----|----|----|----|----|
| 3.5           | TMR@FMR=1%, 10% | 42.88, 88.84 | 64.93, 92.32 | 80.59, 97.63 | 0.094, 0.78 | 0.087, 0.85 | 0.080, 0.83 | 0.079, 0.81 | 0.074, 0.87 | 8.74 | 11.66 |
|               | minDCF ($\tau = 0.001, 0.011$) | 51.26, 89.84 | 68.15, 95.02 | 73.63, 96.97 | 0.092, 0.92 | 0.098, 0.98 | 0.096, 0.96 | 0.078, 0.78 | 0.142 | 16.49 |
| 3.0           | TMR@FMR=1%, 10% | 42.88, 88.84 | 64.93, 92.32 | 80.59, 97.63 | 0.094, 0.78 | 0.087, 0.85 | 0.080, 0.83 | 0.079, 0.81 | 0.074, 0.87 | 8.74 | 11.66 |
|               | minDCF ($\tau = 0.001, 0.011$) | 51.26, 89.84 | 68.15, 95.02 | 73.63, 96.97 | 0.092, 0.92 | 0.098, 0.98 | 0.096, 0.96 | 0.078, 0.78 | 0.142 | 16.49 |
| 2.5           | TMR@FMR=1%, 10% | 42.88, 88.84 | 64.93, 92.32 | 80.59, 97.63 | 0.094, 0.78 | 0.087, 0.85 | 0.080, 0.83 | 0.079, 0.81 | 0.074, 0.87 | 8.74 | 11.66 |
|               | minDCF ($\tau = 0.001, 0.011$) | 51.26, 89.84 | 68.15, 95.02 | 73.63, 96.97 | 0.092, 0.92 | 0.098, 0.98 | 0.096, 0.96 | 0.078, 0.78 | 0.142 | 16.49 |

- **TMR@FMR=1%, 10%** is increased by $\{0.23%, 0.97\%\}$, EER is reduced by 0.99%, and minDCF($\tau = 0.001, 0.011$) is reduced by $\{0.026, 0.02\}$.
- In all the four speaker verification experiments (Experiments 2 to 5) on the degraded Fisher dataset given in Table 2 and Figure 4, 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 noisy environments (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 the presence of babble noise, compared to the F-16 noise, is observed to be significantly lower. This indicates speech babble as one of the more disruptive speech degradations for speaker recognition tasks. All the algorithms trained on DeepVOX features, as compared to MFCC, LPC or MFCC-LPC features, gain significant performance improvements.
- On an average across the seven speaker verification experiments (Experiments 6 to 12), all the algorithms gain performance benefits when the MFCC, LPC and MFCC-LPC features are replaced with DeepVOX features for training the models. Replacing the best performing baseline feature (MFCC-LPC) by DeepVOX features in the 1D-Triplet-CNN-online algorithm improves the verification performance (TMR@FMR=1%, 10%) by $\{14.72%, 8.7\%\}$, reduces the EER by 3.67%, and minDCF($\tau = 0.001, 0.011$) by $\{0.009, 0.11\}$. Similarly, for the 1D-Triplet-CNN, xVector-PLDA, and iVector-PLDA algorithm, the DeepVOX features improve speaker verification performance over the best performing baseline feature (MFCC-LPC). The 1D-Triplet-CNN(DeepVOX) method also outperforms the RawNet2 across all the performance metrics. The TMR@FMR=1%, 10% is increased by $\{25.32%, 17.62\%\}$, EER is reduced by 10.21%, and minDCF($\tau = 0.001, 0.011$) is reduced by $\{0.004, 0.14\}$. Furthermore, the proposed method’s performance benefits compared to the RawNet2 is even greater in the cross-noise experiments (Experiments 3 and 5), demonstrating its superior resilience to mismatched degraded audio conditions.
Figure 4. DET curves for the speaker verification experiments on the VOXCeleb2 (Exp. 1), degraded Fisher (Exp. 2 to 5, the clean and degraded NIST SRE 2008, 2010, and 2018 datasets (Exp. 6 to 12), and the multi-lingual subset of NIST SRE 2008 dataset (Exp. 13 to 15) using RawNet2, iVector-PLDA, xVector-PLDA, 1D-Triplet-CNN, and 1D-Triplet-CNN-online algorithms on MFCC, LPC, MFCC-LPC, and DeepVOX feature sets.

Figure 5. (a) TMR@FMR=1% and (b) EER under varying audio length on the clean NIST SRE 2008 dataset. 1D-Triplet-CNN(DeepVOX) performs the best across varying lengths of test audio.

and 12) on the degraded NIST SRE 2008 dataset.

- In the three speaker verification experiments (Experiments 13 to 15, given in Table 4) on multi-lingual speakers from the NIST SRE 2008 dataset, DeepVOX features perform the best across all the algorithms and metrics. The incorporation of DeepVOX features, compared to the MFCC-LPC features, in the 1D-Triplet-CNN-online algorithm, improves the verification performance (TMR@FMR={1%, 10%}) by {23.95%, 8.97%}, reduces the EER by 4.99% and minDCF(P_{tar} = \{0.001, 0.01\}) by \{0.02, 0.21\}. Similar performance benefits of DeepVOX features were noted for the 1D-Triplet-CNN, xVector-PLDA, and iVector-PLDA algorithms, as well. The 1D-Triplet-CNN(DeepVOX) method also outperforms the RawNet2 across all the performance metrics. The TMR@FMR={1%, 10%} is increased by \{10.18%, 5.19\}, EER is reduced by 3.24%, and minDCF(P_{tar} = \{0.001, 0.01\}) is reduced by \{0.019, 0.12\}.

- It is interesting to note the effect of language on verification performance in Experiments 13 to 15. Best speaker verification performance is achieved in Experiment 13, where the models are trained on English speech data and evaluated on same-language English-only speech audio pairs. However, introduction of same-language multi-lingual speech audio pairs to the evaluation set (in Experiment 14) reduces the verification performance (TMR@FMR=1%) of 1D-Triplet-CNN-online by 3.70% for the DeepVOX features, 14.28% for the MFCC-LPC features, 4.11% for the MFCC features, and 17.46% for the LPC features. Furthermore, re-evaluating the same models on cross-language multi-lingual speech audio pairs in Experiment 15 results in the largest reduction in verification performance, verifying the impact...
of language-familiarity effect [35] in all algorithms and features used in our experiments. It is important to note that the detrimental effects of the language-familiarity effect (in Experiment 14) are observed to be the weakest at 22.49% (performance reduction (TMR@FMR=1%)) for the DeepVOX features compared to 40.76% for the MFCC-LPC features, 39.31% for the MFCC features, and 37.91% for the LPC features, using the best-performing 1D-Triplet-CNN-online algorithm.

- In the experimental results given in Table 3 and illustrated in Figure 5, we notice a gradual decrease in verification performance (across all algorithms and features) with the decrease in length of audio samples in the testing data. However, the loss in performance is observed to be much lower with the usage of DeepVOX features compared to MFCC, LPC, or MFCC-LPC features across all the algorithms. The 1D-Triplet-CNN-online algorithm using DeepVOX features suffers a performance (TMR@FMR=10%) reduction of 10%, compared to a reduction of 32% using MFCC-LPC features, 45% using MFCC features, 34% using LPC features, when the audio length is reduced from 3.5 seconds to 0.5 seconds. Similar trends were observed for the 1D-Triplet-CNN, xVector-PLDA, and the iVector-PLDA algorithms across the DeepVOX, MFCC-LPC, MFCC, and LPC features. For the RawNet2 algorithm, a performance loss of 48% is observed when the length of raw input audio is reduced from 3.5 seconds to 0.5 seconds. It is important to note that when compared to the 1D-Triplet-CNN based algorithms, relatively larger performance losses are observed for the iVector-PLDA, xVector-PLDA, and RawNet2 algorithms, across all the features. However, using the DeepVOX features improves the robustness of even the iVector-PLDA and xVector-PLDA algorithms when performing speaker verification on speech samples of limited duration, thereby asserting the effectiveness of the DeepVOX features in the task.

6 ABLATION STUDY OF DEEVOX

In this section, we use ‘Guided Backpropagation’ [52] to analyze the type of speech information being extracted by the DeepVOX feature. Such an analysis reveals 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 diverse speakers and recording conditions in the training data. For evaluation, we choose audio samples from the TIMIT [20] dataset due to the availability of ground-truth information for analysis of frequency sub-bands essential for speaker recognition [21], [30]. 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, we extract speech utterances corresponding to the five phonemes /ah/, /eh/, /iy/, /ow/, /uw/ from a randomly chosen speaker in the TIMIT dataset. The speech audio of these phonemes is then fed to the trained DeepVOX model to extract corresponding DeepVOX features and subsequently extract the corresponding relevance signals. The input speech signal and the corresponding mean relevance signal are then compared using the Praat [12] toolkit (see Figure 5). While the waveform representation of the original input signal and the corresponding mean relevance signal differ visually, pitch contour analysis of the signals reveals that the relevance signal successfully captures the F0 information from the input speech signal. 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 [28].

Operational Frequency-range of DeepVOX: Similar to [38], we plot the input audio signal (in red color) and corresponding relevance signal (in blue color) on the Power Spectral Density (PSD) plots (given in Figure 7). The PSD plots are inspected for frequency-band overlap in the input audio signal and the corresponding mean relevance signal. The overlap indicates the frequency components of the input audio signal that DeepVOX captures for performing speaker recognition. As observed in Figure 7(a), the trained DeepVOX model reliably models a clean speech input signal in the frequency range of 0 to 4000Hz, with better modeling performance observed in the range of 2000Hz to 4000Hz, which is known to contain highly discriminative speaker-dependent information [21], [30]. This demonstrates DeepVOX’s ability to use spectral information in the frequency range of 0 to 4000Hz for performing speaker recognition.

Effect of Audio Degradation on the DeepVOX: Furthermore, as shown in Figure 7(b), the trained DeepVOX model also

![Figure 6](image-url) A visual comparison of the waveform (top row) and F0 contour (bottom row) for the /ah/ phoneme and its corresponding relevance signal obtained for the DeepVOX model, using the Praat toolkit. Similar results were observed for the /eh/, /iy/, /ow/, and /uw/ phonemes.
reliably models a speech signal degraded with synthetic car noise from the NOISEX-92 dataset [55]. However, it fails to model the synthetic car noise in absence of speech, as shown in Figure 7 [(c)]. This demonstrates DeepVOX’s ability to selectively model the speech components and reject the background noise in an audio sample for performing speaker recognition.

Layer-wise magnitude frequency response of the DeepVOX: Finally, we also plotted (see Figure 8) the layer-wise cumulative magnitude frequency response of the convolution filters in the DeepVOX model trained on the VoxCeleb2 dataset. Here we observed that while the initial three layers behave as a multi-band pass filter, the later layers act as low-pass filters. Specifically, the first three layers’ cumulative magnitude frequency response shows peaks in the frequency range of 0-800Hz and 1500-3000Hz. Comparing to the acoustic characteristics of the human voice in American English [25], the first peak (0-800Hz) is specifically suited for capturing the fundamental frequency (F0) and first formant (F1) of the human voice (the average F0 is 195Hz and average F1 is 595Hz) and the second peak (1500-3000Hz) can capture the second (F2) and third (F3) formants of the human voice (the average F2 is 1734Hz and the average F3 is 2826Hz). Therefore, the initial layers of the DeepVOX model learn to capture important speaker-dependent speech characteristics (F0, F1, F2, and F3) from input speech audio and are well-suited for application in a speaker recognition system.

7 APPLICATIONS OF DEEPVOX

Speaker recognition systems find applications in several different domains, including telephone banking [6], E-Commerce [9] and forensics [8], and personal virtual assistants [4] in the form of voice-controlled user interfaces. However, with the increase in the applications of speaker verification technology, its surface area for incoming threats of circumvention or misuse is also increased. For example, the authors in [55] developed an adversarial audio sample that can be played to stop Amazon Alexa from being activated, thus launching a form of denial-of-service attack (DoS attack). Another form of attack called voice spoofing [32] can be used to impersonate a target user and fraudulently gain access to sensitive user data. Recently, with the advent of DeepFake technology, it is now possible to synthesize realistic speech audio or create convincing alterations of existing speech audios in the public domain to cause widespread panic and confusion [53]. In such scenarios, a robust speaker recognition system, such as DeepVOX, can help thwart the attempts of unauthorized users to illegitimately access a voice interface through voice spoofing or launching DOS attacks.

Towards that end, in this work, we specifically evaluated DeepVOX’s generalizability across a wide variety of speech audio, ranging from telephonic speech conversations in the Fisher Speech Corpora to the nearly-unconstrained interview speeches from the VOXCeleb dataset. Furthermore, we also explored a wide variety of speech audio degradations and assessed their impact on the speaker verification performance of DeepVOX-based models. We specifically introduced the experiments with a mismatch in the audio degradations in the train and evaluation sets (Experiments 3 and 5 in Table 2) and mismatch in spoken language in experiments 14 and 15 in Table 2 to simulate the effect of domain mismatch on the speaker verification performance. While we do notice a performance drop in case of domain mismatch across all the methods and feature combinations tested in this work, the negative impact of domain mismatch is notably reduced across all the scenarios when the DeepVOX replaces traditional speech features (MFCC and LPC). This demonstrates that the DeepVOX in its current form is relatively robust to the adverse effects of mismatch between the training and testing conditions and can be applied to a wide variety of applications discussed above, where domain mismatch is expected. Furthermore, we believe it is possible to tweak the DeepVOX hyperparameters such as the number of DeepVOX filterbanks, the type, length, and stride of the windowing function to adapt the DeepVOX method to specific datasets and audio conditions and further improve its performance.

Additionally, from an implementation perspective, it is important to note that we designed the DeepVOX to offer either an end-to-end learnable or a drop-in replacement for handcrafted filterbanks such as MFCC. For example, on the one hand, we can train DeepVOX end-to-end with any speaker embedding extraction system, as our experiments do with the 1D-Triplet-CNN system. On the other hand, we can replace a fixed MFCC-based feature extraction pipeline with a pre-trained DeepVOX filterbank to asynchronously train a speech embedding method, as shown in our experiments with the xVector and iVector based methods. Also, note that DeepVOX features are generated at the frame level like traditional MFCC features and it should not be confused with a fixed-dimensional speech embedding extractor such as xVector. Therefore, in summary, DeepVOX provides a learnable a time-domain speech filter bank that can either be used to train robust end-to-end speaker recognition systems from scratch or retrofit into existing speaker recognition frameworks. Furthermore, the DeepVOX-based speaker recognition system’s robustness to various non-ideal audio conditions, such as background noise, language mismatch, and short audio duration, makes it an essential tool in the arsenal of digital audio forensics to protect and verify the integrity of data before using it.

8 CONCLUSION

The performance of short-term speech feature extraction techniques, such as MFCC, is dependent on the design of handcrafted filterbanks such as the Mel filterbank. While such techniques are easy to use and do not require any training data, they do not adapt well to diverse non-ideal audio conditions. Therefore, it is beneficial to develop feature extraction techniques, such as DeepVOX, that can robust across diverse non-ideal audio conditions, as evident in the experimental results. The frequency analysis of the learned DeepVOX filterbanks indicates that it 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 rare cases such as Experiment 8 in Table 2, where certain combinations of noise characteristics in the training and testing sets create challenging scenarios for the proposed DeepVOX. Therefore, it is important to continue research in the development of feature extraction algorithms that builds upon and improves the currently proposed algorithm. Furthermore, as discussed in Section 5.1.1, the DeepVOX algorithm has a limitation of being trained only 200 audio frames at a time; hence, it cannot benefit from training on longer audio samples. We plan to extend our DeepVOX model by incorporating methods for automatically learning from audio samples of varying lengths.

Acknowledgment

We thank Dr. Shantanu Chakrabarty and Dr. Kenji Aono from Washington University in St. Louis for providing us their audio degradation tool, used in this work. We also thank Dr. Joseph P. Campbell from MIT Lincoln Lab for the useful discussions.

References

[1] 2008 NIST speaker recognition evaluation set part 2. LDC2011S07. https://catalog.ldc.upenn.edu/LDC2011S05. Accessed: 2018-03-06.
[2] 2010 NIST speaker recognition evaluation test set LDC2010S06. https://catalog.ldc.upenn.edu/LDC2010S06. Accessed: 2018-03-06.
[3] 2013 NIST speaker recognition evaluation test set LDC2013S04. https://catalog.ldc.upenn.edu/LDC2013S04. Accessed: 2018-02-27.
[4] Amazon Alexa voice recognition. https://www.theverge.com/circuitbreaker/2017/10/11/1646620/20-amazon-echo-multi-user-voice-new-feature. Accessed: 2017-12-29.
[5] Google home voice recognition. https://www.cnet.com/news/ai-google-home-good-at-voice-recognition/. Accessed: 2017-12-29.
[6] HSBC voice id making telephone banking safer than ever. https://www.hsbc.co.uk/1/2/voice-id. Accessed: 2017-12-29.
[7] MATLAB voice activity detection by spectral energy. https://github.com/JarvusChen/MATLAB-Voice-Activity-Detection-by-Spectral-Energy. Accessed: 2018-03-06.
[8] Morpho and agnitio partner, bring voice biometrics to protect payments. https://findbiometrics.com/morpho-and-agnitio-partner-bring-voice-biometrics-to-criminal-id-214610. Accessed: 2018-06-13.
[9] Voicevault biometrics to protect payments. https://findbiometrics.com/voicevault-biometrics-to-protect-payments-25131/ Accessed: 2018-06-13.
[10] Wellsfargo voice verification. https://www.wellsfargo.com/privacy-security/voice-verification/. Accessed: 2017-12-29.
[11] Y. Belghiti, J. Lourdour, R. Collobert, and J. Weston. Curriculum learning. In Proceedings of the 26th Annual International Conference on Machine Learning, 2009.
[12] P. Boersma et al. Praat, a system for doing phonetics by computer. Glot international, 5, 2002.
[13] A. Chowdhury and A. Ross. Extracting sub-glottal and supra-glottal features from MFCC using convolutional neural networks for speaker identification in degraded audio signals. In JOBL, IEEE, 2017.
[14] A. Chowdhury and A. Ross. Fusing MFCC and LPC features using 1D Triplet CNN for speaker recognition in severely degraded audio signals. IEEE Transactions on Information Forensics and Security, 2020.
[15] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. arXiv:1806.05622, 2018.
[16] C. Cieri, D. Miller, and K. Walker. Fisher English training speech parts 1 and 2. Philadelphia: Linguistic Data Consortium, 2004.
[17] N. Dekah, P. Kenny, R. Dekah, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing, 19, 2011.
[18] C. Espy-Wilson, S. Munocha, and S. Vishnuhota. 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. Amrouch. Consolidating product spectrum and gammatone filterbank for robust speaker verification under noisy conditions. In International Conference on Intelligent Systems Design and Applications (ISDA), 2015.
[20] W. Fisher, G. Doddington, and K. Goudie-Marshall. The DARPA speech recognition research database: specifications and status. In Proc. DARPA Workshop on Speech Recognition, 1986.
[21] L. F. Gallardo, M. Wagner, and S. Möller. Spectral sub-band analysis of speaker verification employing narrowband and wideband speech. In Odyssey, Citeseer, 2014.
[22] D. Garcia-Romero and C. Espl-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In INTERSPEECH, 2011.
[23] J. Guo, R. Yang, H. Arsikere, and A. Alwan. Robust speaker identification via fusion of subglottal resonances and cepstral features. The Journal of the Acoustical Society of America, 141, 2017.
[24] H. Hermansky. Perceptual linear predictive (plp) analysis of speech. The Journal of the Acoustic Society of America, 87, 1990.
[25] J. Hillenbrand, L. A. Getty, M. J. Clark, and K. Wheeler. Acoustic characteristics of American english vowels. The Journal of the Acoustical Society of America, 97, 1995.
[26] J.-w. Jung, S.-h. Kim, H.-j. Shim, J.-h. Kim, and H.-J. Yu. Improved rawnet with feature map scaling for text-independent speaker verification using raw waveforms. INTERSPEECH, 2020.
[27] A. Kanagasundaram, R. Vogt, D. B. Dean, S. Sridharan, and M. W. Mason. I-vector based speaker recognition on short utterances. In Proceedings of the 12th Annual Conference of the International Speech Communication Association, 2011.
[28] Y. 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.
[29] C. Kim and R. M. Stern. Power-normalized cepstral coefficients (pncc) for robust speech recognition. IEEE/ACM Transactions on audio, speech, and language processing, 2016.
[30] T. Kinnunen. Spectral features for automatic text-independent speaker recognition. Licentiate’s thesis, 2003.
[31] T. Kinnunen and H. Li. An overview of text-independent speaker recognition: From features to supervectors. Speech Communication, 52, 2010.
[32] T. Kinnunen, Z.-Z. Wu, K. A. Lee, F. Sedlak, E. S. Chng, and H. Li. Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4401–4404, 2012.
[33] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter. Self-normalizing neural networks. arXiv:1706.02515, 2017.
[34] 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:1705.0204, 2017.
[35] J. Li, S. Qu, X. Li, J. Szary, J. Z. Kolter, and F. Metze. Adversarial music: Real world audio adversary against wake-word detection system. In Advances in Neural Information Processing Systems, pages 11908–11918, 2019.
[36] L. Lu, Y. Dong, X. Zhao, J. Liu, and H. Wang. The effect of language factors for robust speaker recognition. In IEEE ICASSP, 2009.
[37] B. Milner and X. Shao. Speech reconstruction from mel-frequency cepstral coefficients using a source-filter model. In INTERSPEECH, 2002.
[38] H. Muckenhirn, V. Abrol, M. M. Doss, and S. Marcel. Understanding and visualizing raw waveform-based CNNs. In INTERSPEECH, 2019.
[39] H. Muckenhirn, M. M. Doss, and S. Marcel. Towards directly modeling raw speech signal for speaker verification using CNNs. In IEEE ICASSP, 2018.
[40] L. Muda, M. Begam, and I. Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. arXiv preprint arXiv:1003.4085, 2010.
[41] K. Murty and B. Yegnanarayana. Combining evidence from residual phase and MFCC features for speaker recognition. Signal Processing Letters, 13, 2006.
[42] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative normalizing neural networks. arXiv:1706.02515, 2017.
[43] 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 International Conference on Biometrics (ICB), 2019.
[44] S. Sadjadi and J. Hansen. Hilbert envelope based features for robust speaker identification under reverberant mismatched conditions. In IEEE ICASSP, 2011.
[45] S. Sadjadi, M. Slaney, and L. Heck. MSR identity toolbox v1.0: A MATLAB toolbox for speaker-recognition research. Speech and
[49] J. Schatzman. Accuracy of the discrete fourier transform and the fast fourier transform. *Journal on Scientific Computing*, 17, 1996.

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

[51] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur. X-vectors: Robust DNN embeddings for speaker recognition. In *IEEE ICASSP*, 2018.

[52] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv:1412.6806*, 2014.

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

[54] S. Umesh, L. Cohen, and D. Nelson. Fitting the mel scale. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 1, pages 217–220. IEEE, 1999.

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

[56] 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*. IEEE, 2001.

[57] X. Zhao, Y. Shao, and D. Wang. Casa-based robust speaker identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 2012.

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

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