# Optimization of data-driven filterbank for automatic speaker verification

Susanta Sarangi, Md Sahidullah, Goutam Saha

# Optimization of data-driven filterbank for automatic speaker verification

Susanta Sarangi[a,*], Md Sahidullah[b], Goutam Saha[a]

[a] *Department of Electronics & Electrical Communication Engineering, Indian Institute of Technology, India-721302, Kharagpur, India*
[b] *Université de Lorraine, CNRS, Inria, LORIA, F-54000, Nancy, France*

**Abstract**

Most of the speech processing applications use triangular filters spaced in mel-scale for feature extraction. In this paper, we propose a new data-driven filter design method which optimizes filter parameters from a given speech data. First, we introduce a frame-selection based approach for developing speech-signal-based frequency warping scale. Then, we propose a new method for computing the filter frequency responses by using *principal component analysis* (PCA). The main advantage of the proposed method over the recently introduced deep learning based methods is that it requires very limited amount of unlabeled speech-data. We demonstrate that the proposed filterbank has more speaker discriminative power than commonly used mel filterbank as well as existing data-driven filterbank. We conduct *automatic speaker verification* (ASV) experiments with different corpora using various classifier back-ends. We show that the acoustic features created with proposed filterbank are better than existing *mel-frequency cepstral coefficients* (MFCCs) and *speech-signal-based frequency cepstral coefficients* (SFCCs) in most cases. In the experiments with VoxCeleb1 and popular i-vector back-end, we observe 9.75% relative improvement in equal error rate (EER) over MFCCs. Similarly, the relative improvement is 4.43% with recently introduced x-vector system. We obtain further improvement using fusion of the proposed method with standard MFCC-based approach.

*Keywords:* Mel scale, Frequency warping function, Pitch, Speech-signal-based scale, Principal component analysis (PCA), NIST speaker recognition evaluation (SRE), VoxCeleb1,

## 1. Introduction

Speech is a short-term stationary signal [1] which contains information related to the spoken content, speaker's identity, speaker's emotion, spoken language, etc. The speaker recognition technology recognizes persons from their speech [2]. The *automatic*

*Corresponding author
*Email addresses:* sksarangi@ece.iitkgp.ac.in (Susanta Sarangi), md.sahidullah@inria.fr (Md Sahidullah), gsaha@ece.iitkgp.ernet.in (Goutam Saha)

*speaker verification* (ASV) is one of the important tasks in speaker recognition where two voice signals are compared by machine for deciding whether they are produced by the same speaker or not. The ASV technology finds its application in voice biometrics for authentication tasks in both logical and physical access scenarios [3, 4] and also help in the judicial system to compare an unknown speaker's voice with a known suspect's voice [5, 6]. The performance of the ASV system is reliable in *controlled conditions*; however, in the real-world situations, the performance is considerably degraded due to the variations in *intrinsic factors* (speaker's emotion, health, age, etc.) and *extrinsic factors* (background noise, channel, room impulse response, etc.) [7]. To achieve a good performance in practical applications, the ASV system should be robust against these unwanted variations.

A typical ASV system consists of three main modules: *frame-level feature extractor*, *segment-level feature (embedding) extractor*, and *classifier*. The frame-level feature extraction unit converts raw speech waveform into a sequence of acoustic feature vectors [2, 8]. Most of the ASV studies use short-term spectral features which are based on the knowledge of speech production and perception model. Some studies use high-level features as complementary information which represent other speaking characteristics such as, speaking rate and pronunciation style [9]. The classifier module further parameterizes the features into statistical models [2]. For efficient use of the ASV systems in different real-world applications, we need a feature extraction method which should be robust to unwanted variations in the speech signal and computationally inexpensive [2]. Improving the robustness of acoustic feature usually reduces the effort from classifier for improving the ASV system performance. The scope of this work is limited to the development of a new robust feature extraction algorithm for real-world applications.

Among all the existing cepstral features, *mel-frequency cepstral coefficients* (MFCCs) are the most popular and widely used for the ASV as well as other speech processing tasks such as automatic speech recognition [10], speaker diarization [11], spoofing countermeasures [12], etc. The recently introduced *x-vector* based ASV system, which drew attention in previous NIST speaker recognition evaluations [13, 14, 15], also uses MFCCs as acoustic features. The MFCC computation process involves mel scale integration followed by logarithmic compression and *discrete cosine transform* (DCT). The MFCCs are very popular for the following reasons. First, the computation process utilizes mel filterbank analysis, which is partially inspired by the processing of the audio signal by the human auditory system. Second, the computation process involves *fast Fourier transform* (FFT) and matrix multiplication which makes it more computationally efficient compared to other methods such as *linear prediction cepstral coefficients* (LPCCs) or *line spectral frequencies* (LSFs) [16]. Third, MFCCs are also suitable for different feature-level compensation methods such as *relative spectral* (RASTA) processing [17], *cepstral mean and variance normalization* (CMVN), and *feature warping* [18]. Though the MFCCs are relatively more robust compared to other cepstral features such as *linear frequency cepstral coefficients* (LFCCs) or LPCCs, the ASV performance with MFCCs are severely degraded in real-world conditions due to the mismatch of acoustic conditions in enrollment (or speaker registration) and verification (or speaker authentication) phase [19, 20]. To overcome some of the shortcomings of MFCCs, various acoustic features like *frequency domain linear prediction* (FDLP) [21], *cochlear frequency cepstral coefficients* (CFCCs) [22], *power-normalized cepstral coefficients* (PNCCs) [23], *mean Hilbert envelope coefficients* (MHECs) [24], *Gammatone frequency cepstral coeffi-*

*cients* (GFCCs) [25], *constant-Q cepstral coefficients* (CQCCs) [26], *time-varying linear prediction* (TVLP) [27], and *locally-normalized cepstral coefficients* (LNCCs) [28] were proposed. All these features even though achieve better performance in noisy condition, they require a large number of user-defined parameters. These parameters further need to be manually tuned for different environmental conditions. The overall process seems to be difficult for a system-developer. Also, improving feature robustness beyond a certain level is extremely difficult, especially for a wide range of degradation [21, 24]. Besides, most of those features are also computationally more expensive than MFCCs. The MFCCs, on the other hand, have lesser number of free parameters. This study develops a data-driven feature extraction method which follows the same principle as MFCC but derives the parameters from the speech data itself. Unlike the feature extraction methods discussed before, which require "hand-crafted" parameters, the feature extraction method with parameters computed in a *data-driven* procedure reduces the effort needed for manual fine-tuning. The data-driven methods also show the improvement in robustness when large corpora are used in training strong discriminative models [29].

The data-driven acoustic feature extraction methods use speech data to compute the parameters of the feature extraction algorithm. We classify those methods into two broad categories. One of them uses discriminative approaches such as the *artificial neural network* (ANN) or *linear discriminant analysis* (LDA). These methods require labeled speech data. The other type does not apply the discriminative approach but utilizes some speech science knowledge during parameter estimation. In other words, they learn the feature extraction parameters directly from the speech data without using any class label information. Some of the popular data-driven speech feature extraction methods are discussed in Table 1. Most of the methods are discriminative in nature, and they are generally investigated for automatic speech recognition (ASR) tasks. In ASV research, data-driven feature extraction methods have drawn relatively less attention [30].

In this work, we perform detailed analysis of a data-driven feature extraction method for ASV which utilizes only audio-data for computing the desired parameters, in contrast to most of the data-driven techniques that require additional metadata such as speech (*e.g.*, phoneme) or speaker information. We select *speech-signal-based frequency cepstral coefficient* (SFCC), and this feature has demonstrated promising performance in speech and speaker recognition applications [20, 45]. The method is also very similar to MFCC; however, in contrast to MFCC which applies handcrafted mel scale, SFCC utilizes a *frequency warping* scale that is computed by a data-driven approach. Since the filterbank parameters are computed prior to the feature extraction step, its effective computational time is same as that of MFCCs, and thus considerably lower than other recently proposed features such as FDLP, MHEC or CQCC. The current study extends our preliminary study [45] which introduced the basic data-driven frequency warping [20] in speaker recognition. In this work, we further improve this method by optimizing the scale and by computing the other parameters in a data-driven manner. By performing separability analysis with F-ratio, we have demonstrated that the proposed features are more speaker discriminative than standard MFCCs. Our ASV experiments conducted with different ASV systems agree with this analysis. The major contributions of this work are summarized below.

- We improve the basic data-driven scale with frame selection. With comprehensive analysis and experimental results, we demonstrate that selective use of speech-

Table 1: Selected works on data-driven feature extraction methods for various speech applications (ASR: Automatic speech recognition, ASV: Automatic speaker verification, SAD: Speech activity detection).

| **Work** | **Methodology** | **Task** |
|---|---|---|
| [31] | Neural network is trained with speech features of larger temporal context and used to create data-driven features called TempoRAl Patterns (TRAPs). | ASR |
| [32] | This work investigates data-driven temporal filter with oriented principal component analysis (OPCA) that reduces channel variability. | ASV |
| [33] | The filterbank is derived from phonetically labeled speech data using LDA. | ASR |
| [34] | Data-driven LDA is applied on the logarithmic critical-band power spectrum of speech. | ASR |
| [35] | This method uses TRAP followed by TANDEM. The TRAP estimator provides multiple evidences in terms of posterior probabilities from frequency-localized overlapping time-frequency regions of speech signal computed with the help of data-driven transformation of contextual information. Next, TANDEM converts the frequency-localized evidences to features. | ASR |
| [36] | Data-driven temporal filters are designed using PCA, LDA and minimum classification error (MCE) framework. | ASR |
| [37] | Speech segments are created using a data-driven and automatic language independent speech processing (ALISP). | ASV |
| [38] | This work uses F-ratio to adjust the center and edge frequencies of the filterbank and the F-ratio is computed for speaker separability. | ASV |
| [20] | Data-driven frequency warping is obtained by dividing the long-term average spectrum (LTAS) into subbands of equal energies. | ASR |
| [39] | A multi-layer perceptron (MLP) is trained to classify speech and non-speech frames. The outputs of the MLP are used as posterior features. | SAD |
| [40] | Combination of convolutional neural network (CNN) and long short-term memory (LSTM) is used to learn neural network parameters to classify the context-dependent state labels. | ASR |
| [41] | CNN is used to learn time-domain filter parameters and the network is trained to classify the context-dependent state labels. | ASR |
| [42] | The filterbank is learned in an unsupervised manner using convolutional restricted Boltzmann machine (ConvRBM) with clean and noisy audio data. | ASR |
| [43] | The triangular mel filter is approximated using Gaussian function and the parameters of this pseudo filter are learned using DNN. | ASR |
| [44] | Computational steps of mel-frequency spectral coefficients (MFSCs) are implemented with neural network where the parameters are learned using convolution layers with a goal of maximizing phone recognition accuracy. | ASR |
| [30] | A CNN-based architecture SincNet is introduced which learns the lower and upper cut-off frequencies of the subband filters. Each filter is approximated with the help of a pair of Sinc functions in time-domain and its parameters are tuned by maximizing speaker classification accuracy. | ASV |

frames helps to create more reliable frequency warping scale.

- We introduce a data-driven way for computing filter responses as an alternative to the auditory motivated triangular filters. Our proposed method computes the filterbank response in an unsupervised way with a smaller amount of speech data in contrast to the discriminative approaches that require class labels and a larger amount of speech data.

- We evaluate the proposed features with a state-of-the-art x-vector based ASV sys-

tem which currently utilizes either MFCCs or log-mel energy features.

The rest of the paper is organized as follows. Section 2 explains the baseline cepstral feature extraction methods for both mel and data-driven scale. The next section presents the proposed method for improving the data-driven scale. We propose the data-driven approach of computing filter responses in Section 4. We discuss the experimental setup in Section 5, and we show the results in Section 6. Finally, we conclude in Section 7 with a discussion on limitations of this study and possible future directions.

## 2. Cepstral features based on filterbank

A general block diagram of cepstral feature extraction methods using a filterbank is shown in Fig. 1. After pre-processing steps such as framing and windowing, the short-term power spectrum of speech frames is multiplied with a filterbank frequency response. Then, cepstral features are created by performing DCT on log-energies of filterbank output. In MFCC computation, we place the triangular-shaped filters in the mel scale. However, for SFCCs, triangular filters are placed in data-driven speech-signal-based scale. In the following sub-sections, we briefly describe these two feature extraction methods.

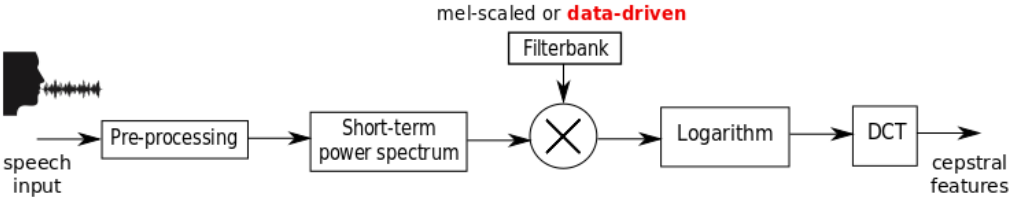


Figure 1: Block diagram of a typical cepstral feature extraction method.

### *2.1. MFCCs: fixed mel scale*

The MFCC feature extraction scheme introduced in [46] provides a straightforward way to compute cepstral features. Since then, it had been the state-of-the-art in different speech-based applications including speaker recognition [47]. It uses the mel scale [48] based triangular filterbank for the creation of cepstral features. There are several alternatives to mel scale representations [49]. The most commonly used equation to convert linear frequency $f$ to mel frequency $f_{mel}$ is

$$f_{mel} = 2595 \log_{10} \left(1 + \frac{f}{700}\right). \tag{1}$$

In the mel scale domain, the frequency axis is divided into equidistant points. By considering those points as filter edge-frequencies, the filters are placed by keeping 50% overlap with the adjacent one [50]. In the MFCC computation step, the pre-emphasized speech signal is first segmented into frames of 20-30 ms typically with an overlap of 50%. After that, we perform the short-time Fourier transform (STFT) of the speech frames.

Then, we compute filterbank energies by using mel filterbank. Finally, DCT is performed over logarithm of filterbank energies to get cepstral features. The detailed procedure to compute MFCCs can be found in [19, 51].

*2.2. SFCCs: data-driven scale*

The SFCC is a data-driven cepstral feature extraction method which computes the non-linear scale from the training speech. This scale was initially proposed for speech recognition [20] and later has been successfully applied in speaker recognition [45]. The SFCC extraction method replaces the mel scale with a data-driven scale, and the rest of the process is the same as the MFCC computation process. The following paragraph describes the steps required to get data-driven scale.

The scale computation involves the computation of long-term average power spectrum (LTAS) of speech data. The LTAS per speech utterance is computed first by averaging the short-term power spectrum over all the frames in the utterance. Then, average LTAS is computed over all the speech utterances present in a corpus for computating the scale. In the next step, the logarithm of LTAS is divided into equal area intervals to compute the filter edge frequencies.

The derivation of the speech-based data-driven scale is described in the following steps.

**1. Computation of LTAS**: Let $u$ be a speech utterance of $N_l$ frames. Its LTAS is expressed as,

$$P[k] = \frac{1}{N_l} \sum_{i=1}^{N_l} X_i[k], \tag{2}$$

where $X_i[k]$ is the energy spectrum and $k$ is the index of frequency bin.

**2. Computation of average LTAS**: The average LTAS is computed as the ensemble average of LTAS of all speech utterances in a corpus, and it is defined as,

$$\bar{P}[k] = \frac{1}{N_s} \sum_{i=1}^{N_s} P[k], \tag{3}$$

where $N_s$ is the total number of speech utterances in a corpus.

**3. Computation of cumulative log power spectrum**: Now, if we want to place $Q$ filters, we divide the $\log \bar{P}[k]$ into frequency subbands of $Q$ equal areas. We compute the area of the $j$-th band as,

$$A_j = \sum_{k=k_l^j}^{k_h^j} \log \bar{P}[k], \tag{4}$$

where $j = 1, 2, 3, ..., Q$. Here $k_l^j$ and $k_h^j$ are the lower and upper band for $j$-th filter and they are selected in such a manner that $A_1 = A_2 = A_3 = \ldots = A_Q$. We also consider the lower edge frequency of the first filter as 0 Hz and the higher edge frequency of the

last filter as the Nyquist frequency. In practice, it is not possible to get $A_i$s exactly equal in numerical values, and they are made approximately equal.

**4. Computation of warping scale**: Finally, the scale is computed by interpolating the filterbank center frequencies to their mapped values which are obtained with the help of the following equation [20],

$$W\left[\frac{k_l^j + k_h^j}{2}\right] = \frac{j}{Q}, \tag{5}$$

where $j = 1, 2, 3, ..., Q$.

Eq. (5) gives the required frequency points to design filters in the filterbank structure. The cepstral features computed with this scale is referred to as SFCCs. This scale used in SFCC computation is shown in Fig. 3 along with standard mel, ERB, and Bark scale as well as the scale proposed in Section 3.

To compute this scale, we do not require speaker labels for the corpus, unlike most of the methods listed in Table 1. During this scale computation, all the speech frames are used which are selected by a speech activity detection (SAD) method. This includes all types of speech frames showing different spectral characteristics; however, we do not necessarily need the entire speech corpus as LTAS can be obtained with a small subset of available data. In the next section, we consider a frame selection technique to select useful frames for better ASV performance.

## 3. Data-driven frequency warping using selected frames

The frame selection strategy is used in speaker recognition task for fast implementation in real-time application [52, 53]. In this work, we select a subset of speech frames for developing warping scale.

The conventional mel scale is a psychophysical scale for pitch perception. This experimental scale was first formulated with the help of a perceptual test by playing tones of fixed frequency to the listeners [48]. The participants were asked to tune the frequency of another adjustable tone according to the perceived half-value of the fixed tone. All the tones were played at a constant loudness of 60 dB. The scale was formulated by fitting a curve that maps the numerical value of linear frequency to the perceived value.

We note that the mel scale development is originally a subjective method which might be biased to the selected listeners [54]. Therefore, instead of subjective criterion, in data-driven method, we replace human being with the objective criterion of equal energy of voice signal. During this process, we consider all types of speech data irrespective of the voice production mechanism. This crude selection of speech frames include unvoiced speech frames created with random noise passing through a narrow constriction of the vocal tract. This unvoiced frames have no harmonic structure and closely resemble the uniform distribution of noise spectra [1]. Therefore, we propose to select only the voiced frames having pitch information for our data-driven scale formulation process.

Fig. 2 shows the spectrogram and pitch contour of the speech signal, which is taken from the NIST SRE 2001 corpus.

Fig. 3 shows the normalized plot of both auditory and data-driven scales. We observe that the data-driven scales have lower resolution at the both ends of the frequency band,

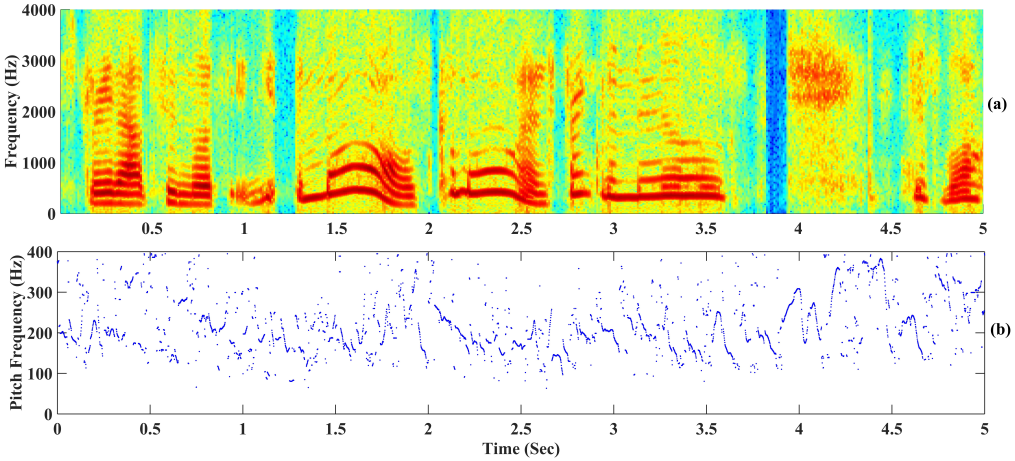


Figure 2: Illustration of (a) spectrogram and (b) pitch contour of a speech signal taken from NIST SRE 2001 speech corpus. We compute the pitch values using *the pitch estimation filter robust to high levels of noise* (PEFAC) method as studied in [55].

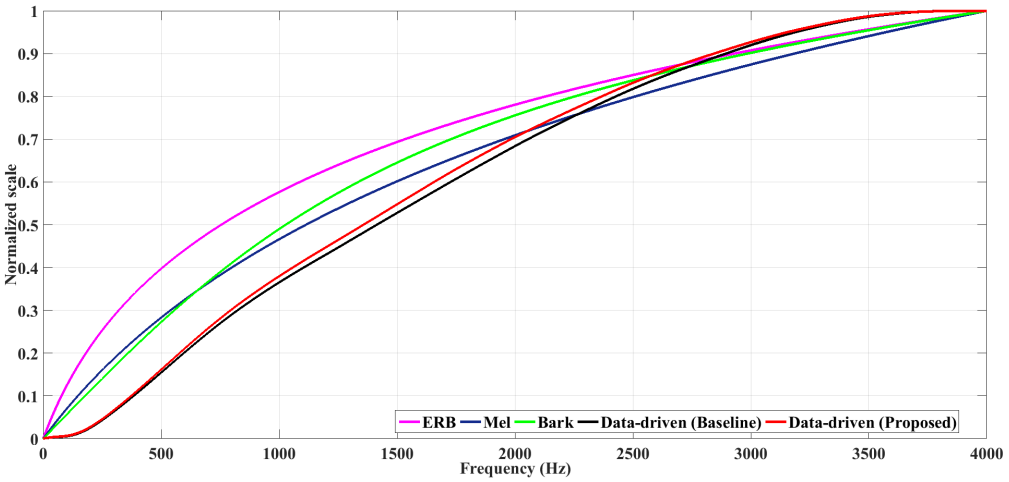


Figure 3: The frequency warping function for Mel, ERB, Bark, SFCC, and proposed scale.

and higher resolution everywhere else. This is expected as the speech files for NIST SRE 2001 are collected over telephone channel with a frequency range of $300 - 3700$ Hz. Therefore, we hypothesize that the filterbank placed according to the newly derived scale will help to capture more relevant information than mel filterbank or standard data-driven filterbank.

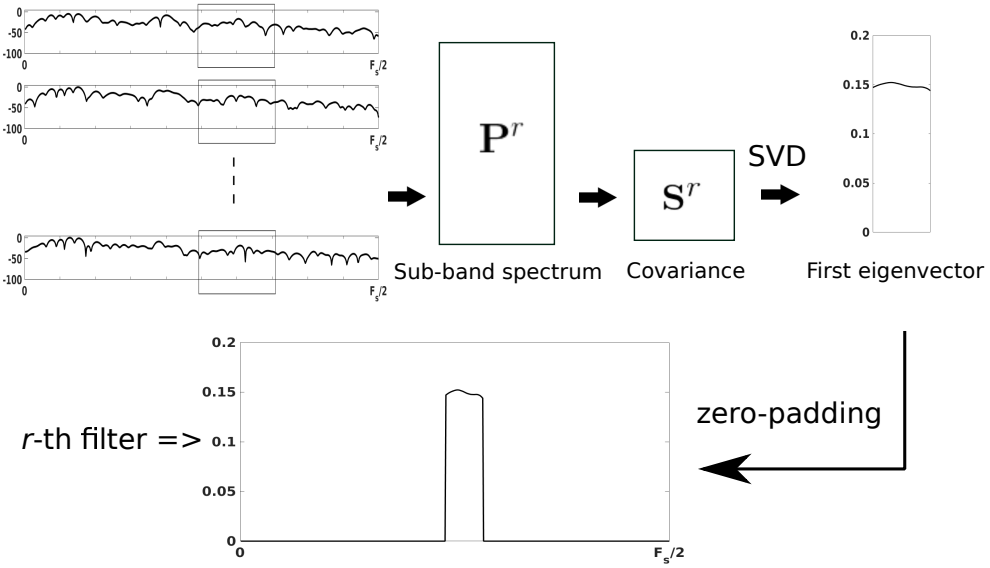


Figure 4: Figure showing proposed data-driven method for computing frequency response of a filter. Here the matrix $\mathbf{P}^r$ contains log power spectrum of all the frames corresponding to the $r$-th subband.

## 4. Computation of data-driven filter shape using PCA

In traditional MFCCs and SFCCs, we use filters with triangular-shaped frequency responses which closely approximate the auditory filters in cochlea. Other shapes like Gaussian [56] and trapezoidal [57] are also used in speech feature extraction process. The shape of the filters in filterbank assigns non-uniform weights to the subband frequencies.

In this work, the idea is to design the subband filter response so that the output of this filter, computed as the energy, will represent the subband frequency components in a most effective way. In other words, we need to reduce the dimension of the subband frequency components to a single data point. We employ *principal component analysis* (PCA) which is appropriate for finding the most "expressive" representation of the data [58]. Previously, PCA is applied to design the data-driven filterbank with mel scale for robust speech recognition [36, 59]. We propose to apply PCA on the log-power spectrum of each frequency band separately for constructing the filters. The PCA basis with highest eigenvalue, known as "first basis", is used to create the filter frequency response. Since speech signal is a highly correlated process [60], the subband covariance matrix will be positive. Hence all the elements of its eigenvector with highest eigenvalue, i.e., the first basis of PCA, will be non-negative according to the *Perron-Frobenius* theorem [61]. The steps to find the filter shape are summarized below:

**1. Computation of subband covariance matrix**: Let $P_i^r[k]$ be the log power spectrum of $k$-th frequency component for $r$-th subband and $i$-th speech frame. Then the subband covariance matrix corresponding to the $r$-th subband is given as:

$$\mathbf{S}^r = \frac{1}{N_f - 1} \sum_{i=1}^{k_r} (P_i^r[k] - \bar{m}[k])(P_i^r[k] - \bar{m}[k])^\top, \tag{6}$$

where $N_f$ is the number of frames, $k_r$ is the number of frequency bins in $r$-th subband and $\bar{m}[k]$ is the mean subband power spectrum given by,

$$\bar{m}[k] = \frac{1}{N_f} \sum_{i=1}^{N_f} P_i^r[k]. \tag{7}$$

**2. Computation of first PCA basis**: We apply *singular value decomposition* (SVD) [62] to compute the PCA basis of subband covariance matrix. Using SVD, we can write,

$$\mathbf{S}^r = \mathbf{U}\mathbf{V}\mathbf{U}^\top, \tag{8}$$

where $\mathbf{U}$ is the $k_r \times k_r$ orthogonal matrix containing eigenvectors of $\mathbf{S}^r$ in each column, and the diagonal elements of the $k_r \times k_r$ matrix $\mathbf{V}$ contain the singular values. The first column of $\mathbf{U}$, i.e., the first principal component is used to create the $r$-th filter. We apply zero-padding to get the filter frequency response for the entire band. The computation of PCA-based filter response is illustrated in Fig. 4.

The filter shape computed in the above process treats all the frequency components within a subband in an identical manner. However, considering the subbands have overlap with the adjacent bands, we apply tapering function to the power spectrum that assigns higher weights to the frequency components near center frequencies and lower weights to the components near edge frequencies. We use *Hamming window* on the power spectrum data before performing PCA. We also normalize the frequency response to make the highest magnitude unity similar to the mel filters. Fig. 5 illustrates the filters for different frequency warping scale.

In order to analyze the separability of different features, we compute F-ratio [63]. For this analysis, we used 131 speakers from POLYCOST corpus [64]. In Table 2, we showed the F-ratio of log-energies of different feature extraction methods with 20 filters. This demonstrates that the proposed methods have more filters that have higher discriminative characteristics. We also showed the average F-ratio which indicates that the proposed features are better than the MFCC for most cases.

## 5. Experimental setup

### *5.1. Corpora for experiments*

We evaluated our proposed method in NIST (SRE 2001 and SRE 2002) and VoxCeleb (VoxCeleb1) speech corpora [65, 66, 7]. In addition, we evaluate the performance in noisy conditions. Initially, we conducted experiments on NIST SRE 2001 corpus to optimize different parameters. Then, we used those parameters to evaluate the ASV system in the NIST SRE 2002 corpus for both clean and noisy conditions.

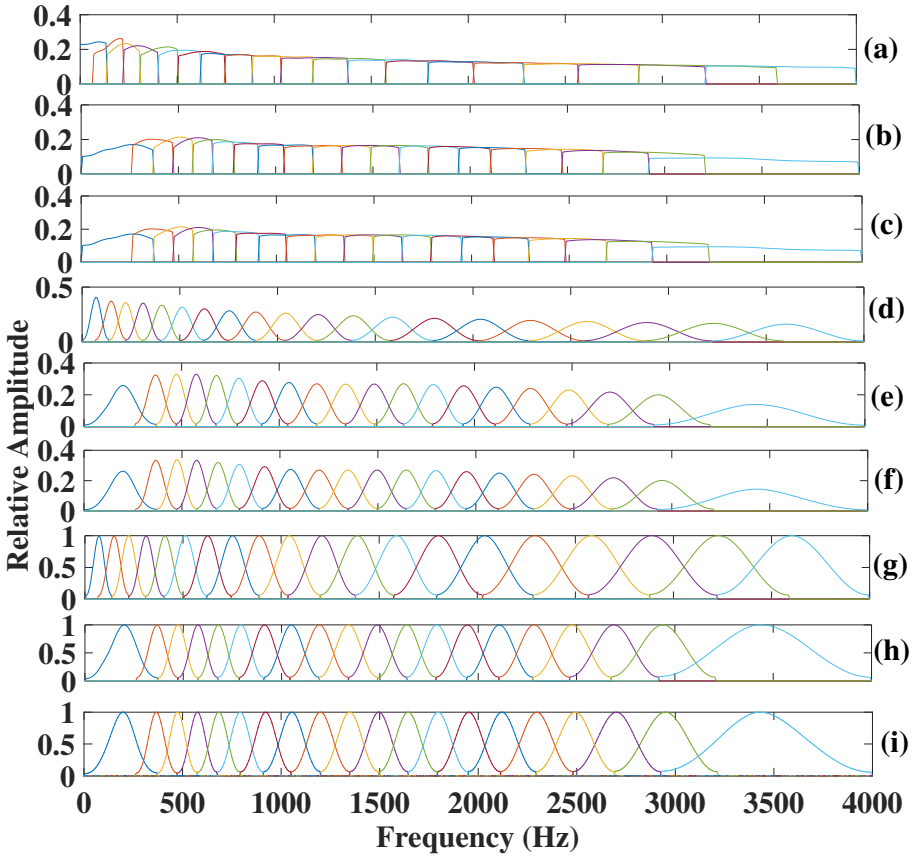


Figure 5: Data-driven filterbank frequency responses using PCA. The filters are shown for different scales: (a) mel, (b) speech-based, and (c) speech-based with pitch. The next three (d, e and f) shows the filter shapes for three scales when Hamming window is applied on the log-power spectrum. The last three (g, h, and i) are for normalized frequency responses. In all cases, the filters are derived from the development set of NIST SRE 2001 corpus.

We use VoxCeleb1 corpus consisting large number of speakers for real-world conditions [7]. This corpus consists of voices of 1251 celebrities collected from the YouTube videos. Out of them, 40 speakers are used for evaluation purpose. The sampling rate of each utterance is of 16 kHz, and average utterance length is 8 seconds. The corpora used in our experiments are summarized in Table 3.

We use the development data for scale computation. The same data is used to train the model parameters and hyper-parameters, i.e., for computing the parameters for UBM, PLDA and T-matrix when required. The enrollment and the test sentences are corrupted with noises where SNRs range from 0 to 40 dB and type of noise is randomly chosen from five noises (white, pink, babble, volvo and factory). We took noise files from NOISEX-92 corpus.

Table 2: F-ratios of log-energies for MFCC features and for three kinds of SFCC features denoted by M1, M2 and M3. M1 indicates the baseline SFCC feature where the scale is computed with all the speech frames. M2 indicates SFCC features when the scale is computed taking speech frames having pitch using pitch estimation algorithm. Finally, M3 indicates the SFCC features when the scale is same as M2 but triangular filters are replaced with window-based PCA filters. The last row shows the average ratio for all the cases.

| Filter No. | MFCC | SFCC | | |
|---|---|---|---|---|
| | | M1 | M2 | M3 |
| 1 | **0.5677** | 0.4107 | 0.4103 | 0.4149 |
| 2 | **0.4241** | 0.3270 | 0.3262 | 0.3281 |
| 3 | **0.3417** | 0.2864 | 0.2862 | 0.2864 |
| 4 | 0.2403 | 0.2855 | 0.2856 | **0.2860** |
| 5 | 0.2015 | 0.2965 | **0.2968** | 0.2961 |
| 6 | 0.2135 | 0.3018 | **0.3022** | 0.3012 |
| 7 | 0.2521 | 0.3209 | **0.3217** | 0.3209 |
| 8 | 0.2607 | 0.3405 | **0.3412** | 0.3410 |
| 9 | 0.2870 | 0.3564 | **0.3571** | 0.3565 |
| 10 | 0.3088 | 0.3773 | **0.3780** | 0.3774 |
| 11 | 0.3252 | **0.3758** | 0.3753 | 0.3749 |
| 12 | 0.3407 | 0.3775 | **0.3785** | 0.3781 |
| 13 | 0.3281 | 0.4093 | **0.4110** | 0.4110 |
| 14 | 0.3511 | 0.4396 | **0.4408** | 0.4404 |
| 15 | 0.3966 | 0.4596 | **0.4598** | 0.4593 |
| 16 | 0.4280 | **0.4469** | 0.4460 | 0.4454 |
| 17 | 0.4160 | 0.4477 | **0.4487** | 0.4484 |
| 18 | 0.4557 | 0.4842 | 0.4853 | **0.4861** |
| 19 | 0.4990 | 0.5164 | **0.5176** | 0.5173 |
| 20 | 0.5696 | 0.5746 | 0.5757 | **0.5845** |
| Avg. | 0.3604 | 0.3917 | 0.3922 | **0.3927** |

### *5.2. Feature extraction*

We extracted the acoustic features from speech frames of 20 ms with 10 ms overlap. For experiments with GMM-UBM and i-vector system, we used 20 filters. We extracted 19 coefficients after discarding the first coefficient. Finally, a 57-dimensional feature vector [67] is formulated after appending delta and double-delta coefficients. The MFCCs are filtered with RASTA processing [17] to remove slowly varying channel effect. Finally, we perform cepstral mean and variance normalization (CMVN) after applying bi-Gaussian modelling based SAD [19]. We use identical pre-processing and post-processing steps for all the features.

### *5.3. Classifier details*

We use three different ASV systems: GMM-UBM, i-vector and DNN-based x-vector. First, we use simple GMM-UBM classifier for conducting experiments with NIST SRE 2001 and NIST SRE 2002 corpora. Then, we evaluate our proposed feature on VoxCeleb1 corpus using i-vector and x-vector system. In order to make the work self-contained, we briefly describe all the classifiers as follows.

Table 3: Summary of the corpora for speaker verification experiments.

| Corpus | No. of speakers | Target models | Test segments | Total trials | True trials | Impostor trials |
|---|---|---|---|---|---|---|
| NIST SRE 2001 | 174 | 174 | 2038 | 22418 | 2038 | 20380 |
| NIST SRE 2002 | 330 | 330 | 3570 | 39270 | 2983 | 36287 |
| VoxCeleb1 | 40 | 4715 | 4713 | 37720 | 18860 | 18860 |

*5.3.1. GMM-UBM system*

In the GMM-UBM system, the feature distribution of the target speakers and the cohort models are represented with a mixture of Gaussian densities [68]. The cohort model, also known as universal background model (UBM) in this context, is trained with several hours of speech data. The UBM is represented as $\lambda_{\text{ubm}} = \{w_i, \mu_i, \mathbf{\Sigma}_i\}_{i=1}^{C}$ where $C$ is the number of Gaussian mixture components, and $w_i$ is the weight, $\mu_i$ is the mean, and $\mathbf{\Sigma}_i$ is the covariance matrix of the $i$-th component. The parameter $w_i$ satisfies the constrain $\sum_{i=1}^{C} w_i = 1$. The enrollment speech model ($\lambda_{\text{enroll}}$) are derived from the UBM using *maximum-a-posteriori* (MAP) adaptation with the target speaker's feature.

During test, we calculate ASV score of the test utterance, $\mathbf{X}_{\text{test}} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ as the log-likelihood ratio (LLR), given by the following equation:

$$\Lambda_{\text{GMM-UBM}}(\mathbf{X}_{\text{test}}, \lambda_{\text{enroll}}) = \log P(\mathbf{X}_{\text{test}}|\lambda_{\text{enroll}}) - \log P(\mathbf{X}_{\text{test}}|\lambda_{\text{ubm}}). \tag{9}$$

Finally, if the ASV score is greater than or equal to a decision threshold, $\theta$, the test speech is considered as spoken by the correct speaker, otherwise an imposter.

In our experiments, we use the development section of NIST SRE 2001 corpus, which consists of six hours of speech data, to train gender-independent UBM of 512 mixture components. We use 10 iterations of *expectation-maximization* (EM) algorithm to estimate the UBM parameters. The target speaker models are created by adapting only the mean vectors of UBM with relevance factor 14.

*5.3.2. i-vector system*

In i-vector method, the GMM concatenated means of the adapted GMM, known as GMM-*supervector*, is projected into a low dimensional space called as total variability (TV) space [69] as,

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w}, \tag{10}$$

where $\mathbf{T}$, $\mathbf{m}$ and $\mathbf{M}$ are the low-rank total variability matrix, the speaker and channel independent supervector (taken from UBM supervector) and the GMM supervector representation of the speech utterance, respectively. Here the $\mathbf{w}$ is called as i-vectors. In order to compute the i-vector representation of a speech utterance, $\mathbf{X}_{\text{utt}}$, we estimate the posterior mean of the i-vector given the centered first-order Baum-Welch statistics as,

$$\mathbf{w}_{\text{utt}} = (\mathbf{I} + \mathbf{T}^{\top}\mathbf{\Sigma}^{-1}\mathbf{N}\mathbf{T})^{-1}\mathbf{T}^{\top}\mathbf{\Sigma}^{-1}\mathbf{F}, \tag{11}$$

where $\mathbf{N}$ is matrix consisting of the zero-order Baum-Welch statistics as the diagonal elements; $\mathbf{F}$ is a vector whose elements are first-order Baum-Welch statistics; and $\mathbf{\Sigma}$ is the residual variability, commonly created from the UBM covariances.

The extracted i-vectors contain channel information. In order to compensate the effect of channel, *probabilistic linear discriminant analysis* (PLDA) is used to compute the similarity between i-vectors of enrollment and test [70]. We use Gaussian PLDA (GPLDA) in our experiment which models the within-class covariance by a full-rank matrix.

The ASV score using PLDA is computed as the likelihood score given as,

$$\Lambda_{\text{PLDA}}(\mathbf{w}_{\text{enroll}}, \mathbf{w}_{\text{test}}) = \log \frac{p(\mathbf{w}_{\text{enroll}}, \mathbf{w}_{\text{test}}|H_s)}{p(\mathbf{w}_{\text{enroll}}|H_d)p(\mathbf{w}_{\text{test}}|H_d)}, \tag{12}$$

where $\mathbf{w}_{\text{enroll}}$ and $\mathbf{w}_{\text{test}}$ are correspondingly the i-vectors of enrollment and test sentences. Here $H_s$ and $H_d$ represent two hypotheses whether two i-vectors are from the same speaker ($H_s$) or not ($H_d$).

In our experiment with i-vector system, we have randomly selected $20,000$ and $50,000$ speech files from VoxCeleb1 dev set for training the UBM and T-matrix, respectively. The PLDA is trained with entire dev set consisting $148,642$ files from 1211 speakers. We also apply linear discriminant analysis (LDA) to improve the speaker discriminativeness of i-vectors with the same data as used in PLDA training. We fix the number of mixture components to 512 and i-vector dimension to 400. The i-vectors are projected to 250 dimensions using LDA. We perform *whitening* and *length normalization* on i-vectors before training GPLDA with 200 dimensional speaker subspace.

*5.3.3. x-vector system*

The x-vector system uses deep neural network to learn the speech representation in a supervised manner unlike the unsupervised linear method used in i-vector approach [47]. The neural network consists of *time-delay neural network* (TDNN) along with statistical pooling followed by fully connected layers. This architecture captures information from a large temporal context from the frame-level speech feature sequences [71]. The TDNN is a fixed-size *convolutional neural network* (CNN) that share weights along the temporal dimension and it is regarded as 1D convolution (Conv1D) or temporal convolution [72]. The x-vector system is trained for speaker classification task at segment level. Finally, the x-vectors are computed from the output of the first fully connected layer.

In our x-vector system implementation, we use five TDNN layers and three fully connected layers as used in [47]. The details of the neural network configuration is shown in Table 4.

We implemented the x-vector system with Python library Keras [73] using TensorFlow [74] as backend. We use *rectified linear unit* (ReLU) [75] and *batch normalization* [76] for all the five TDNN and two fully connected layers. We apply dropout with probability 0.05 only on the two fully connected layers. The parameters of the neural network are initialized with Xavier normal method [77]. The neural network is trained using Adam optimizer [78] with learning rate 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and without weight decay. We train the neural network using speech segments of 1 seconds. We use 20-dimensional MFCCs computed with 20 filters. The MFCCs after dropping non-speech

Table 4: Description of the layers in x-vector architecture.

| **Layer** | **Details** |
|---|---|
| TDNN-1 | Conv1D (#filter = 256, kernel size = 5, dilation rate =1) |
| TDNN-2 | Conv1D (#filter = 256, kernel size = 3, dilation rate =2) |
| TDNN-3 | Conv1D (#filter = 256, kernel size = 3, dilation rate =3) |
| TDNN-4 | Conv1D (#filter = 256, kernel size = 1, dilation rate =1) |
| TDNN-5 | Conv1D (#filter = 1024, kernel size = 1, dilation rate =1) |
| Statistics pooling | Computation of mean and standard deviation |
| FC1 | Fully connected layer (256 nodes) |
| FC2 | Fully connected layer (256 nodes) |
| Softmax | Softmax layer with 1211 outputs |

frames with SAD are processed with utterance-dependent cepstral mean normalization (CMN). The x-vector systems are trained with batch size of 100. We use the minimum validation loss as the stopping criteria. We consider entire VoxCeleb1 dev set consisting 1211 speakers (same data as i-vector extractor training). We used data augmentation as used in standard x-vector system [47]. We extract 256-dimensional embeddings from the fully connected layers (after batch normalization but before applying ReLU).

Table 5: Parameters of the cost function for NIST SREs and VoxCeleb1 corpora.

| Corpus | $C_{miss}$ | $C_{fa}$ | $P_{tar}$ |
|---|---|---|---|
| NIST SRE 2001 and 2002 | 10 | 1 | 0.01 |
| VoxCeleb1 | 1 | 1 | 0.01 |

### *5.4. Performance evaluation*

We evaluate ASV system performance with commonly used evaluation metrics: *equal error rate* (EER) and *minimum detection cost function* (minDCF) computed from the detection error trade-off (DET) curve [2, 79]. The EER is the point in DET curve where the false alarm rate (FAR) and false rejection rate (FRR) are equal. On the other hand, minDCF is computed by formulating a weighted cost function after assigning costs to the error rates followed by minimization of the weighted cost function. The cost function is defined as,

$$C_{det} = C_{miss} \times P_{miss}(\theta) \times P_{tar} + C_{fa} \times P_{fa}(\theta) \times (1 - P_{tar}), \tag{13}$$

where $C_{miss}$ and $C_{fa}$ are the cost of miss and false acceptance, respectively, $P_{miss}(\theta)$ and $P_{fa}(\theta)$ are the probabilities of miss and false acceptance at decision threshold $\theta$, and $P_{tar}$ is the prior target probability. The values of $C_{miss}$, $C_{fa}$ and $P_{tar}$ are chosen according to the evaluation plan of the respective corpus [7, 65, 66] and their values are shown in Table 5.

## 6. Results and discussion

### *6.1. Experiments on NIST SREs with GMM-UBM system*

We evaluate the ASV performances on NIST SREs using GMM-UBM classifier. First, we assess the performance with MFCC and baseline SFCC features for subsequent comparison with the proposed features on NIST SRE 2001 corpus. For SFCC methods, we compute the scale using the development section of NIST SRE 2001 corpus. Table 6 shows the comparison between baseline MFCC, SFCC and proposed one (best one selected among pitch estimation methods mentioned) which indicates that the proposed one performs better than MFCC and SFCC in terms of both the evaluation metrics.

Table 6: Comparison of ASV system performances in terms of EER (in %) and minDCF×100 for MFCC, SFCC, and the proposed features on NIST SRE 2001 corpus using GMM-UBM backend.

| **Feature** | | **EER(in %)** | **minDCF**×100 |
|---|---|---|---|
| MFCC | | 7.70 | 3.39 |
| SFCC (Baseline) | | 7.51 | 3.28 |
| SFCC (Scale with pitch) | [80] | 7.61 | 3.27 |
| | [81] | 7.45 | 3.40 |
| | [82] | 7.31 | **3.23** |
| | [83] | 7.22 | 3.26 |
| | [55] | **7.21** | 3.24 |

We also perform the experiment with data-driven filter shapes created with PCA-based method. In Table 7, we have shown the ASV performance for different scales where the filter is computed with PCA on the development data. We observe that the ASV performance is relatively poor compared to the results with fixed triangular based filters. Interestingly, the proposed scale based features are better than mel scale based features. The pitch based ASV system yields lowest EER amongst all the three systems.

We further apply tapering window (here Hamming) on the subband spectrum before performing PCA. The results are reported in Table 8. We observe noticeable improvement compared to the results of untaperd case in Table 7. Interestingly, the performance obtained with the data-driven filter shapes are sometimes better than the performance with triangular filters. For instance, in case of MFCCs, the minDCFs ($\times$ 100) of the triangular and data-driven filters are 3.39 (Table 6) and 3.35 (Table 8). Similarly, the EER for pitch-based system reduces to 7.11% from 7.21% when windowed and PCA-based data-driven filter is used instead of triangular filters. However, we do not observe improvement in EER with after normalizing the filter response magnitudes though we observe a reduction in cost function values.

We also conduct experiment with NIST SRE 2002 corpus to evaluate the generalization ability of the proposed data-driven approach. In this case, the same development data from the subset of NIST SRE 2001 corpus is used for computing the parameters of data-driven feature extractor. The results are summarized in Table 9. We observe that with triangular filter, mel-scaled filterbank always obtain lower EER and minDCF than the data-driven scale based methods. The reason for this performance is due to domain-mismatch as the scale is computed on the speech files from a different corpus, i.e., NIST SRE 2001. However, we notice that the warping scale based on the selected

Table 7: Comparison of ASV performances with PCA-based data-driven filters using different scales. Results are shown in terms of EER (in %) and minDCF×100 on NIST SRE 2001 corpus using GMM-UBM back-end.

| **Scale** | **EER(in %)** | **minDCF**×100 |
|---|---|---|
| Mel | 8.69 | 3.86 |
| Speech-based | 8.39 | **3.61** |
| Speech-based with pitch | **8.38** | 3.66 |

Table 8: Same as Table 7 but with tapering Window applied on the subbands before applying PCA. We also report the performance with magnitude normalized filters in the last row.

| **Scale** | **EER(in %)** | **minDCF**×100 |
|---|---|---|
| Mel | 7.70 | 3.35 |
| Speech-based | 7.25 | 3.27 |
| Speech-based with pitch | **7.11** | 3.23 |
| Speech-based with pitch (Normalized) | 7.41 | **3.22** |

Table 9: Comparison of ASV performances with fixed (i.e, mel scale with triangular filter) and various data-driven features on NIST SRE 2002 corpus. Performances are shown in terms of EER (in %) and minDCF ×100 using GMM-UBM backend. Here, the scale is computed using development set of NIST SRE 2001 corpus.

| **Filter Shape** | **Scale** | **EER (in %)** | **minDCF** ×100 |
|---|---|---|---|
| Triangular | Mel | 8.76 | 4.07 |
| | Speech-based | 9.15 | 4.45 |
| | Speech-based with pitch | 9.12 | 4.28 |
| PCA | Mel | 9.65 | 4.33 |
| | Speech-based | 9.92 | 4.55 |
| | Speech-based with pitch | 9.96 | 4.57 |
| Window+PCA | Mel | **8.42** | 4.04 |
| | Speech-based | 9.15 | 4.34 |
| | Speech-based with pitch | 8.75 | 4.25 |
| Window+PCA+Norm. | Mel | 8.48 | **4.03** |
| | Speech-based | 9.29 | 4.29 |
| | Speech-based with pitch | 8.91 | 4.33 |

Table 10: Comparison of ASV system performances in noisy conditions. Results are shown in terms of EER (in %) and minDCF×100 on additive noise-corrupted NIST SRE 2002 corpus with GMM-UBM as backend.

| **Methods** | **EER(in %)** | **minDCF**×100 |
|---|---|---|
| MFCC (Baseline) | 18.27 | 8.04 |
| Speech-based with triangular filter | 16.63 | 7.64 |
| Speech-based (pitch) with window & PCA-based filter | **16.02** | **7.56** |

frames from pitch show improvement over the condition where the scale is computed on all the speech frames (*i.e.*, baseline SFCCs). The DET curves of ASV results of selected features are illustrated in Fig. 6 and 7.

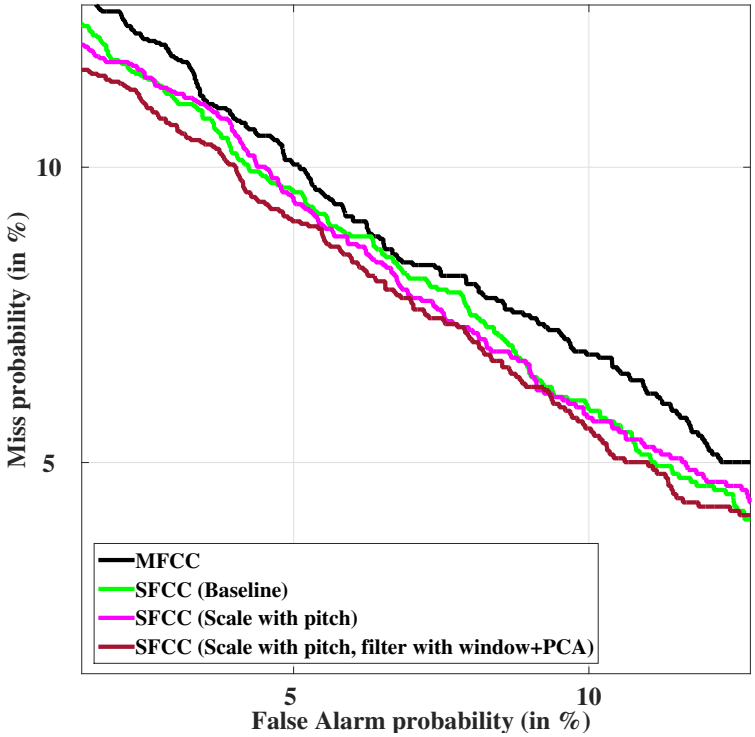


Figure 6: The DET curves ASV system performance using different feature extraction methods on NIST SRE 2001 corpus with GMM-UBM as backend.

Table 11: Comparison of ASV performances on VoxCeleb1 corpus with i-vector system. Results are shown in terms of EER (in %) and minDCF×100 for features based on different scales. The scale is computed on the development set of the VoxCeleb1 corpus.

| **Scale** | **EER(in %)** | **minDCF**×100 |
|---|---|---|
| Mel | 9.95 | 0.747 |
| Speech-based | 9.71 | 0.786 |
| Speech-based with pitch | 9.52 | **0.721** |
| Speech-based (pitch) with window & PCA-based filter | **8.98** | 0.744 |

Table 12: Comparison of ASV performances when in-domain and out of domain (Librispeech and TIMIT) are used for computing scale of data-driven filter. Results are shown in terms of EER (in %) and minDCF×100 on VoxCeleb1 test set using i-vector and PLDA backend.

| **Corpus for scale computation** | **EER(in %)** | **minDCF**×100 |
|---|---|---|
| VoxCeleb1 (in-domain) | **9.52** | **0.721** |
| Librispeech | 10.40 | 0.812 |
| TIMIT | 9.99 | 0.730 |

Even though we do not observe improvement with the data-driven scales, the performances of mel scale based are improved with window and PCA based data-driven filters. We can conclude that scale selection is more sensitive to the corpus selection whereas

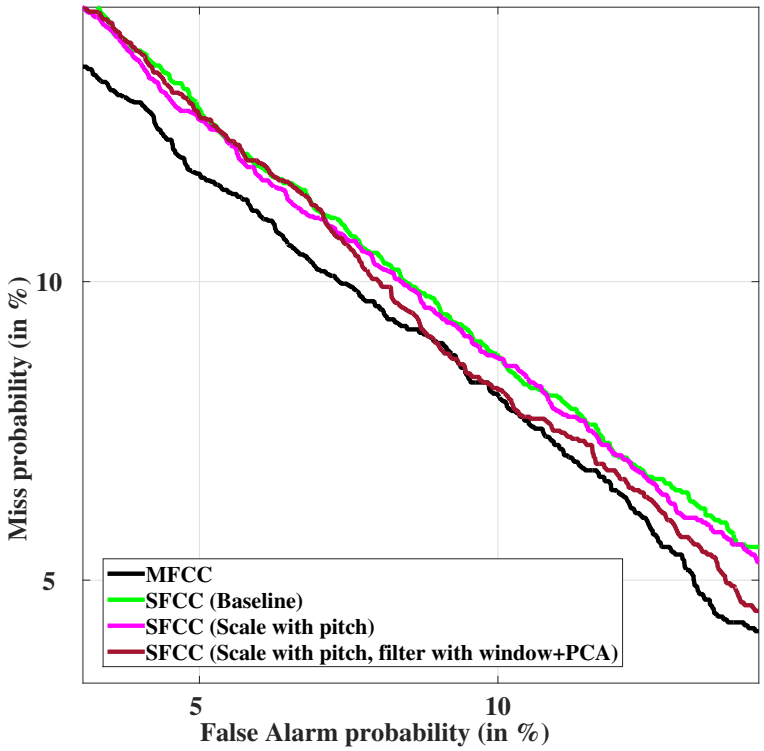


Figure 7: Same as Fig.6 but for NIST SRE 2002 corpus.

filter-responses computed from one dataset generalize well to other datasets.

Finally, the results on noisy conditions are shown in Table 10 and the corresponding DET in Fig. 8. Here, we have found that the propose data-driven features are more robust compared to the baseline MFCCs. The best performance in terms of EER is obtained with data-driven feature where scale is computed from the selected frames with pitch values and the filter shape is computed with windowed spectrum and PCA.

### *6.2. Experiments on VoxCeleb1*

#### *6.2.1. Performance evaluation with i-vector system*

In our experiments with i-vector system on VoxCeleb1, first we compute the scale on the entire development set consisting of 1211 speakers and report the results for different scales in Table 11. We observe that the performance with feature using frame selection based scale yields better performance in terms of both EER and minDCF. We obtain more than 4.30% and 3.48% relative reduction for EER and minDCF, respectively. Fig. 9 shows the DET curve of the ASV system using VoxCeleb1 corpus. From this curve, we find that the proposed features perform better than the other features in ASV task.

In Table 11, the scales are computed with entire development data which is computationally expensive, especially for PCA-based filter shape computation. In the next experiment, we examined the effect of amount of data for scale computation on the performance of ASV system where we chose a small subset of speech utterances from the entire set of 148642 files. We conducted the ASV experiments 10 times where every

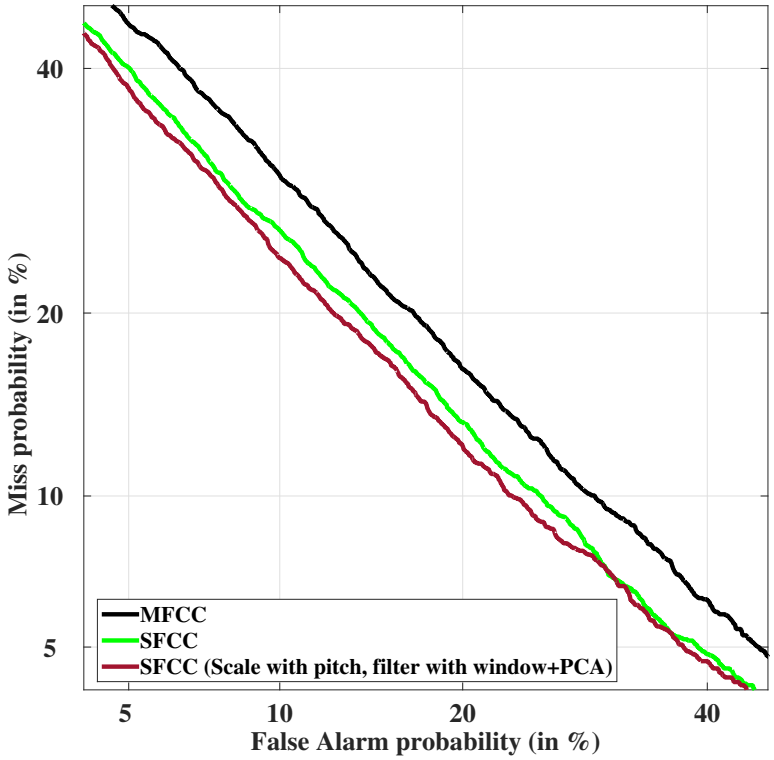


Figure 8: The DET curves of ASV systems based on different feature extraction methods on noise-corrupted version of NIST SRE 2002 corpus using GMM-UBM as backend.

Table 13: Comparison of ASV performances with x-vector representation. Results are shown in terms of EER (in %) and minDCF×100 on VoxCeleb1 test set.

| **Methods** | **Embeddings from FC1** | | **Embeddings from FC2** | |
|---|---|---|---|---|
| | **EER (in %)** | **minDCF**×100 | **EER (in %)** | **minDCF**×100 |
| MFCC | 5.13 | 0.468 | 5.19 | 0.480 |
| Proposed SFCC | 5.03 | 0.468 | 4.96 | 0.502 |
| Score Fusion | **4.45** | **0.421** | **4.56** | **0.446** |

time 0.1% of the speech data are randomly selected. The ASV performances for this randomly chosen small subsets are shown in Fig. 10. The figure also shows the performance with baseline MFCCs and proposed method with full speech data. We observe that filterbank parameters computed with 0.1% of the data shows lower EER than baseline MFCCs. However, we do not observe improvements in minDCF. Interestingly, the ASV performance with 100% of the data for scale computation only gives about 1% relative improvement (in terms of EER) over 0.1% data. To compare the performance with out of domain data, we also conduct experiment where the data for scale formulation is taken from corpus other than VoxCeleb1 in-domain data. We took speech data from Librispeech [84] and TIMIT [85] for this purpose. We use the same VoxCeleb1 data for computing parameters of UBM, T-matrix, LDA and PLDA. The results reported in Ta-

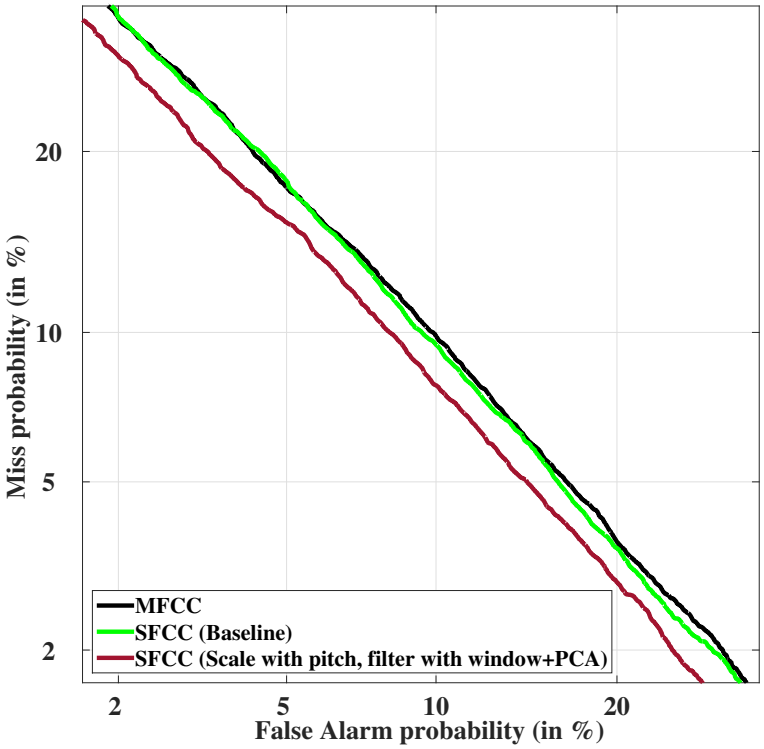


Figure 9: The DET curves ASV systems based on different data-driven feature extraction methods on VoxCeleb1 corpus with i-vector and PLDA-based scoring as backend.

ble 12 indicates that out of domain data considerably degrades the ASV performances. We conclude that the proposed method should be applicable where limited in-domain data is available.

*6.2.2. Performance evaluation with x-vector system*

For experiments with x-vector system, we chose baseline MFCCs and the proposed data-driven feature extraction in which the warping scale and the filter parameters are computed with development data from the VoxCeleb1 corpus. The results of x-vector system with PLDA scoring are summarized in Table 13. In the state-of-the-art x-vector system also, the proposed features are better than conventionally used MFCCs in terms of EER. We showed the results when the embeddings are computed from the output of FC1 and FC2. The improvement is observed for both cases. We did not find improvement in terms of minDCF; however, the proposed features are better than MFCCs in most of the operating points as shown in the DET curve in Fig.11.

Finally, we perform experiments with fused system where scores of MFCC and proposed SFCC are combined with equal weights [86]. The performance is substantially improved with fusion. In EER, we obtained relative improvement of 13.26% and 12.14% over baseline MFCC, respectively for x-vector embeddings computed from FC1 and FC2. This confirms the complementarity of proposed data-driven filterbank with mel filter-

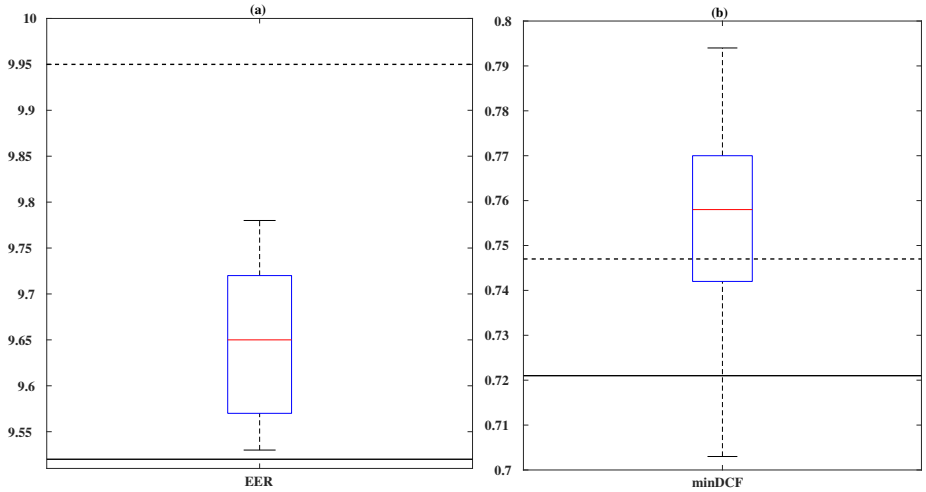


Figure 10: Error bar plot showing ASV performance on VoxCeleb1 where 0.1% of the total speech data are randomly selected for computing filterbank parameters. The results are shown on VoxCeleb1 corpus with i-vector back-end. The dotted horizontal line indicates the performance with baseline MFCCs and continuous horizontal line denotes the performance with proposed method where 100% speech data are used for computing the filterbank parameters.

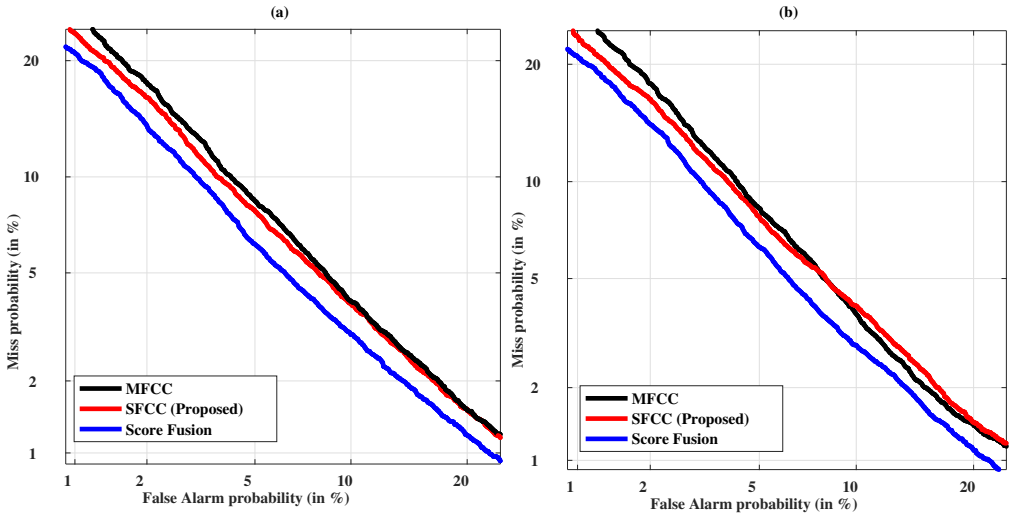


Figure 11: The DET curves of results of ASV system among MFCC, Proposed feature extraction methods and score level fusion on VoxCeleb1 corpus using x-vector system where (a) using FC1 and (b) using FC2.

bank.

## 7. Conclusion

The filterbanks in most of the acoustic feature extraction modules are either handcrafted with some auditory knowledge or learned over a large dataset with some objectives. In this work, we proposed to compute the MFCC filterbank in a data-driven way. We improved the data-driven frequency warping scale by considering voiced frames having pitch information. We demonstrated the superiority of the newly designed warping scale for ASV tasks. We also computed frequency responses of the filters in a data-driven manner from the subband power spectrum using PCA. We showed that both these schemes reduce the speaker recognition error rates. We observed improvements in both matched and mismatch conditions. The proposed feature extraction method is compatible with the state-of-the-art x-vector systems and shows improvement over MFCC-based ASV systems. The proposed method is computationally less expensive than DNN-based data-driven methods. Also, it computes the filterbank parameters (*i.e.*, filter edge frequencies & frequency response) with a small amount of speech data without additional metadata as opposed to the supervised methods which require a large amount of labeled data. We further improved the ASV performance by simple score fusion with an MFCC-based system.

Even though the acoustic features computed with the proposed data-driven filters show improvement over MFCCs, the performance of the proposed features substantially degrades if in-domain audio-data is not available. However, domain-mismatch remains an open challenge for other data-driven feature extractors, too. In future, we plan to explore the data-augmentation methods for addressing the domain mismatch issue. We can compute the filterbank from the augmented speech data and observe its robustness. The objective of this work was not to optimize the number of filters and the amount of overlaps with the adjacent filters. The present work can also be extended in that direction. In this work, we develop the filterbank in a task-independent manner but its application is limited to ASV in the current study. We also plan to adopt the proposed data-driven filterbank for other potential speech processing tasks, such as language and emotion recognition.

## Acknowledgments

The authors would like to express their sincere thanks to the anonymous reviewers and the editors for their valuable comments and suggestions which greatly improved the work in quality and content. The work of Md Sahidullah is supported by Region Grand Est, France. Experiments presented in this paper were partially carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see `https://www.grid5000.fr`).

## References

[1] T. F. Quatieri, Discrete-time speech signal processing: principles and practice, Pearson Education India, 2006.

[2] T. Kinnunen, H. Li, An overview of text-independent speaker recognition: from features to supervectors, Speech Communication 52 (1) (2010) 12–40.

[3] M. Sahidullah, H. Delgado, M. Todisco, T. Kinnunen, N. Evans, J. Yamagishi, K. A. Lee, Introduction to voice presentation attack detection and recent advances, in: Handbook of Biometric Anti-Spoofing, Springer, 2019, pp. 321–361.
[4] J. A. Markowitz, Voice biometrics, Communications of the ACM 43 (9) (2000) 66–73.
[5] B. K. Dumas, Voice identification in a criminal law context, American Speech 65 (4) (1990) 341–348.
[6] J. P. Campbell, W. Shen, W. M. Campbell, R. Schwartz, J. F. Bonastre, D. Matrouf, Forensic speaker recognition, IEEE Signal Processing Magazine 26 (2) (2009).
[7] A. Nagrani, J. S. Chung, A. Zisserman, VoxCeleb: a large-scale speaker identification dataset, in: Proc. INTERSPEECH, 2017, pp. 2616–2620.
[8] J. P. Campbell, Speaker recognition: a tutorial, Proceedings of the IEEE 85 (9) (1997) 1437–1462.
[9] B. Ayoub, K. Jamal, Z. Arsalane, Gammatone frequency cepstral coefficients for speaker identification over VoIP networks, in: Proc. 2016 International Conference on Information Technology for Organizations Development (IT4OD), IEEE, 2016, pp. 1–5.
[10] M. Benzeghiba, R. De Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, et al., Automatic speech recognition and speech variability: a review, Speech Communication 49 (10-11) (2007) 763–786.
[11] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, O. Vinyals, Speaker diarization: a review of recent research, IEEE Transactions on Audio, Speech, and Language Processing 20 (2) (2012) 356–370.
[12] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, H. Li, Spoofing and countermeasures for speaker verification: a survey, Speech Communication 66 (2015) 130–153.
[13] The NIST year 2018 speaker recognition evaluation plan, `https://www.nist.gov/itl/iad/mig/nist-2018-speaker-recognition-evaluation` (2018).
[14] K. A. Lee, et al., I4U submission to NIST SRE 2018: leveraging from a decade of shared experiences, arXiv preprint arXiv:1904.07386 (2019).
[15] NIST 2019 speaker recognition evaluation, `https://www.nist.gov/itl/iad/mig/nist-2019-speaker-recognition-evaluation` (2019).
[16] L. Rabiner, B. Juang, Fundamentals of Speech Recognition, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
[17] H. Hermansky, N. Morgan, RASTA processing of speech, IEEE Transactions on Speech and Audio Processing 2 (4) (1994) 578–589.
[18] J. Pelecanos, S. Sridharan, Feature warping for robust speaker verification, in: Proc. Odyssey: The Speaker and Language Recognition Workshop, 2001, pp. 213–218.
[19] M. Sahidullah, G. Saha, Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition, Speech Communication 54 (4) (2012) 543–565.
[20] K. Paliwal, B. Shannon, J. Lyons, K. Wójcicki, Speech-signal-based frequency warping, IEEE Signal Processing Letters 16 (4) (2009) 319–322.
[21] S. Ganapathy, Signal analysis using autoregressive models of amplitude modulation, Ph.D. thesis, Johns Hopkins University, USA (2012).
[22] Q. Li, Y. Huang, An auditory-based feature extraction algorithm for robust speaker identification under mismatched conditions, IEEE Transactions on Audio, Speech, and Language Processing 19 (6) (2011) 1791–1801.
[23] C. Kim, R. M. Stern, Power-normalized cepstral coefficients (PNCC) for robust speech recognition, IEEE/ACM Transactions on Audio, Speech and Language Processing 24 (7) (2016) 1315–1329.
[24] S. O. Sadjadi, J. H. Hansen, Mean Hilbert envelope coefficients (MHEC) for robust speaker and language identification, Speech Communication 72 (2015) 138–148.
[25] X. Zhao, Y. Shao, D. Wang, CASA-based robust speaker identification, IEEE Transactions on Audio, Speech, and Language Processing 20 (5) (2012) 1608–1616.
[26] M. Todisco, H. Delgado, N. W. Evans, Articulation rate filtering of CQCC features for automatic speaker verification, in: Proc. INTERSPEECH, 2016, pp. 3628–3632.
[27] V. Vestman, D. Gowda, M. Sahidullah, P. Alku, T. Kinnunen, Speaker recognition from whispered speech: a tutorial survey and an application of time-varying linear prediction, Speech Communication 99 (2018) 62–79.
[28] V. Poblete, F. Espic, S. King, R. M. Stern, F. Huenupán, J. Fredes, N. B. Yoma, A perceptually-motivated low-complexity instantaneous linear channel normalization technique applied to speaker verification, Computer Speech & Language 31 (1) (2015) 1–27.
[29] J. H. Hansen, T. Hasan, Speaker recognition by machines and humans: a tutorial review, IEEE Signal Processing Magazine 32 (6) (2015) 74–99.
[30] M. Ravanelli, Y. Bengio, Speaker recognition from raw waveform with SincNet, in: Proc. Spoken

Language Technology Workshop (SLT), IEEE, 2018, pp. 1021–1028.
[31] H. Hermansky, S. Sharma, Temporal patterns (TRAPS) in ASR of noisy speech, in: Proc. ICASSP, Vol. 1, 1999, pp. 289–292.
[32] N. Malayath, H. Hermansky, S. Kajarekar, B. Yegnanarayana, Data-driven temporal filters and alternatives to GMM in speaker verification, Digital Signal Processing 10 (1-3) (2000) 55–74.
[33] L. Burget, H. Hermansky, Data-driven design of filter bank for speech recognition, in: Proc. International Conference on Text, Speech and Dialogue, Springer, 2001, pp. 299–304.
[34] N. Malayath, H. Hermansky, Data-driven spectral basis functions for automatic speech recognition, Speech Communication 40 (4) (2003) 449–466.
[35] H. Hermansky, TRAP-TANDEM: Data-driven extraction of temporal features from speech, in: Proc. IEEE ASRU, 2003, pp. 255–260.
[36] J. W. Hung, L. S. Lee, Optimization of temporal filters for constructing robust features in speech recognition, IEEE Transactions on Audio, Speech, and Language Processing 14 (3) (2006) 808–832.
[37] A. El Hannani, D. T. Toledano, D. Petrovska-Delacrétaz, A. Montero-Asenjo, J. Hennebert, Using data-driven and phonetic units for speaker verification, in: Proc. Odyssey: The Speaker and Language Recognition Workshop, 2006, pp. 1–6.
[38] X. Lu, J. Dang, An investigation of dependencies between frequency components and speaker characteristics for text-independent speaker identification, Speech Communication 50 (4) (2008) 312–322.
[39] S. Thomas, S. H. Mallidi, T. Janu, H. Hermansky, N. Mesgarani, X. Zhou, S. Shamma, T. Ng, B. Zhang, L. Nguyen, M. Spyros, Acoustic and data-driven features for robust speech activity detection, in: Proc. INTERSPEECH, 2012, pp. 1985–1988.
[40] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, O. Vinyals, Learning the speech front-end with raw waveform CLDNNs, in: Proc. INTERSPEECH, 2015, pp. 1–5.
[41] Y. Hoshen, R. J. Weiss, K. W. Wilson, Speech acoustic modeling from raw multichannel waveforms, in: Proc. ICASSP, 2015, pp. 4624–4628.
[42] H. B. Sailor, H. A. Patil, Novel unsupervised auditory filterbank learning using convolutional RBM for speech recognition, IEEE/ACM Transactions on Audio, Speech, and Language Processing 24 (12) (2016) 2341–2353.
[43] H. Seki, K. Yamamoto, S. Nakagawa, A deep neural network integrated with filterbank learning for speech recognition, in: Proc. ICASSP, 2017, pp. 5480–5484.
[44] N. Zeghidour, N. Usunier, I. Kokkinos, T. Schaiz, G. Synnaeve, E. Dupoux, Learning filterbanks from raw speech for phone recognition, in: Proc. ICASSP, 2018, pp. 5509–5513.
[45] S. K. Sarangi, G. Saha, A novel approach in feature level for robust text-independent speaker identification system, in: Proc. Fourth International Conference on Intelligent Human Computer Interaction (IHCI), 2012, pp. 1–5.
[46] S. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, IEEE Transactions on Acoustics, Speech, and Signal Processing 28 (4) (1980) 357–366.
[47] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, S. Khudanpur, X-vectors: robust DNN embeddings for speaker recognition, in: Proc. ICASSP, 2018, pp. 5329–5333.
[48] S. Stevens, J. Volkmann, E. Newman, A scale for the measurement of the psychological magnitude pitch, The Journal of the Acoustical Society of America 8 (3) (1937) 185–190.
[49] T. Ganchev, N. Fakotakis, G. Kokkinakis, Comparative evaluation of various MFCC implementations on the speaker verification task, in: Proc. Tenth International Conference on Speech and Computer (SPECOM), Vol. 1, 2005, pp. 191–194.
[50] S. Chakroborty, A. Roy, G. Saha, Improved closed set text-independent speaker identification by combining MFCC with evidence from flipped filter banks, International Journal of Signal Processing 4 (2) (2007) 114–121.
[51] T. Ganchev, Speaker recognition, Ph.D. thesis, University of Patras, Greece (Nov 2005).
[52] T. Kinnunen, E. Karpov, P. Franti, Real-time speaker identification and verification, IEEE Transactions on Audio, Speech, and Language Processing 14 (1) (2006) 277–288.
[53] G. Sarkar, G. Saha, Real-time implementation of speaker identification system with frame picking algorithm, Procedia Computer Science 2 (2010) 173–180.
[54] D. D. Greenwood, Mailing list from 2009; discussion on bias in the mel scale due to Hysteresis effects, `@http://lists.mcgill.ca/scripts/wa.exe?A2=ind0907d&L=auditory&P=389` (2013).
[55] S. Gonzalez, M. Brookes, PEFAC-a pitch estimation algorithm robust to high levels of noise, IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 22 (2) (2014) 518–530.

[56] S. Chakroborty, G. Saha, Feature selection using singular value decomposition and QR factorization with column pivoting for text-independent speaker identification, Speech Communication 52 (9) (2010) 693–709.
[57] H. Hermansky, Perceptual linear predictive (PLP) analysis of speech, The Journal of the Acoustical Society of America 87 (4) (1990) 1738–1752.
[58] C. M. Bishop, Pattern Recognition and Machine Learning, 1st Edition, Springer, 2006.
[59] S. M. Lee, S. H. Fang, J. W. Hung, L. S. Lee, Improved MFCC feature extraction by PCA-optimized filter-bank for speech recognition, in: Proc. IEEE ASRU, 2001, pp. 49–52.
[60] N. Jayant, P. Noll, Digital Coding of Waveforms: Principles and Applications to Speech and Video, Prentice-Hall, Englewood Cliffs NJ, USA, 1984, p. 688, chapter 8.
[61] C. R. Johnson, R. A. Horn, Matrix analysis, 3rd Edition, Cambridge University Press, 2013, Chapter 8.
[62] G. H. Golub, C. F. Van Loan, Matrix Computations, 3rd Edition, Johns Hopkins University Press, 2012.
[63] S. Nicholson, B. Milner, S. Cox, Evaluating feature set performance using the F-ratio and J-measures, in: Proc. EUROSPEECH, 1997, pp. 413–416.
[64] J. Hennebert, H. Melin, D. Petrovska, D. Genoud, POLYCOST: A telephone-speech database for speaker recognition, Speech Communication 31 (2-3) (2000) 265–270.
[65] The NIST year 2001 speaker recognition evaluation plan, `http://www.itl.nist.gov/iad/mig/tests/spk/2001/2001-spkrec-evalplan-v05.9.pdf` (2001).
[66] The NIST year 2002 speaker recognition evaluation plan, `http://www.itl.nist.gov/iad/mig/tests/spk/2002/2002-spkrec-evalplan-v60.pdf` (2002).
[67] M. Sahidullah, T. Kinnunen, Local spectral variability features for speaker verification, Digital Signal Processing 50 (2016) 1–11.
[68] D. A. Reynolds, T. F. Quatieri, R. B. Dunn, Speaker verification using adapted Gaussian mixture models, Digital Signal Processing 10 (1-3) (2000) 19–41.
[69] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification, IEEE Transactions on Audio, Speech, and Language Processing 19 (4) (2010) 788–798.
[70] P. Rajan, A. Afanasyev, V. Hautamäki, T. Kinnunen, From single to multiple enrollment i-vectors: Practical PLDA scoring variants for speaker verification, Digital Signal Processing 31 (2014) 93–101.
[71] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. J. Lang, Phoneme recognition using time-delay neural networks, IEEE Transactions on Acoustics, Speech, and Signal Processing 37 (3) (1989) 328–339.
[72] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.
[73] F. Chollet, et al., Keras, `https://keras.io` (2015).
[74] M. Abadi, et al., TensorFlow: large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL `https://www.tensorflow.org/`
[75] V. Nair, G. Hinton, Rectified linear units improve restricted Boltzmann machines, in: Proc. ICML, 2010, pp. 807–814.
[76] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proc. ICML, 2015, pp. 448–456.
[77] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proc. of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.
[78] D. P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proc. ICLR, 2015, pp. 1–15.
[79] A. Martin, G. Doddington, T. Kamm, M. Ordowski, M. Przybocki, The DET curve in assessment of detection task performance, in: Proc. EUROSPEECH, 1997, pp. 1895–1898.
[80] T. Drugman, A. Alwan, Joint robust voicing detection and pitch estimation based on residual harmonics, in: Proc. INTERSPEECH, 2011, pp. 1973–1976.
[81] X. Sun, Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio, in: Proc. ICASSP, Vol. 1, 2002, pp. I–333–I–336.
[82] A. De Cheveigné, H. Kawahara, YIN, a fundamental frequency estimator for speech and music, The Journal of the Acoustical Society of America 111 (4) (2002) 1917–1930.
[83] J. Wise, J. Caprio, T. Parks, Maximum likelihood pitch estimation, IEEE Transactions on Acoustics, Speech, and Signal Processing 24 (5) (1976) 418–423.
[84] V. Panayotov, G. Chen, D. Povey, S. Khudanpur, Librispeech: an ASR corpus based on public domain audio books, in: Proc. ICASSP, 2015, pp. 5206–5210.

[85] V. Zue, S. Seneff, J. Glass, Speech database development at MIT: TIMIT and beyond, Speech Communication 9 (4) (1990) 351–356.
[86] V. Hautamäki, T. Kinnunen, F. Sedlák, K. A. Lee, B. Ma, H. Li, Sparse Classifier Fusion for Speaker Verification, IEEE Transactions on Audio, Speech, and Language Processing 21 (8) (2013) 1622–1631.

**Susanta Sarangi** is currently pursuing Ph.D. in area of speech processing from the Department of Electronics & Electrical Communication Engineering, Indian Institute of Technology Kharagpur. He obtained Master of Technology degree in Electronics and Communication Engineering from Biju Patnaik University of Technology, Odisha in 2008 and Bachelors of Engineering in Electronics and Telecommunication Engineering from Utkal University, Odisha in 2002. Before joining Ph.D., he was an Assistant Professor in Institute of Technical Education and Research (ITER), Siksha 'O' Anusandhan Deemed to be University, Odisha. His research interests include speech & audio signal processing, signal processing, and machine learning.

**Md Sahidullah** received his Ph.D. degree in the area of speech processing from the Department of Electronics & Electrical Communication Engineering, Indian Institute of Technology Kharagpur in 2015. Prior to that he obtained the Bachelors of Engineering degree in Electronics and Communication Engineering from Vidyasagar University in 2004 and the Masters of Engineering degree in Computer Science and Engineering from West Bengal University of Technology in 2006. In 2014-2017, he was a postdoctoral researcher with the School of Computing, University of Eastern Finland. In January 2018, he joined MULTISPEECH team, Inria, France as a post-doctoral researcher where he currently holds a starting research position. His research interest includes robust speaker recognition and spoofing countermeasures. He is also part of the organizing team of two Automatic Speaker Verification Spoofing and Countermeasures Challenges: ASVspoof 2017 and ASVspoof 2019. Presently, he is also serving as Associate Editor for the IET Signal Processing and Circuits, Systems, and Signal Processing.

**Goutam Saha** received his B.Tech. and Ph.D. degrees from the Department of Electronics & Electrical Communication Engineering, Indian Institute of Technology (IIT) Kharagpur, India in 1990 and 2000, respectively. In between, he served industry for about four years and obtained a five year fellowship from Council of Scientific & Industrial Research, India. In 2002, he joined IIT Kharagpur as a faculty member where he is currently serving as a Professor. His research interests include analysis of audio and bio signals.