



# PF-Net: Personalized Filter for Speaker Recognition from Raw Waveform

Wencheng Li, Zhenhua Tan<sup>(✉)</sup>, Zhenche Xia, Danke Wu, and Jingyu Ning

School of Software, Northeastern University, Shenyang 110819, China  
{liwencheng, xiazhenche, wudk2019}@stumail.neu.edu.cn,  
{tanzh, ningjy}@mail.neu.edu.cn

**Abstract.** Speaker recognition using i-vector has been replaced by speaker recognition using deep learning. Speaker recognition based on Convolutional Neural Networks (CNNs) has been widely used in recent years, which learn low-level speech representations from raw waveforms. On this basis, a CNN architecture called SincNet proposes a kind of unique convolutional layer, which has achieved band-pass filters. Compared with standard CNNs, SincNet learns the low and high cut-off frequencies of each filter. This paper proposes an improved CNNs architecture called PF-Net, which encourages the first convolutional layer to implement more personalized filters than SincNet. PF-Net parameterizes the frequency domain shape and can realize band-pass filters by learning some deformation points in frequency domain. Compared with standard CNN, PF-Net can learn the characteristics of each filter. Compared with SincNet, PF-Net can learn more characteristic parameters, instead of only low and high cut-off frequencies. This provides a personalized filter bank for different tasks. As a result, our experiments show that the PF-Net converges faster than standard CNN and performs better than SincNet. Our code is available at [github.com/TAN-OpenLab/PF-NET](https://github.com/TAN-OpenLab/PF-NET).

**Keywords:** Speaker recognition · Raw waveform · Personalized filters · Deep learning

## 1 Introduction

In our daily life, we are always receiving and conveying a variety of information from the outside world, and voice information is an important part of it. Audio signal processing plays an important role in the field of artificial intelligence and machine learning. Because each person's voice organs are different, their voices and tones are different. Apart from physical differences, each person has his own unique way of speaking, including using a specific accent, rhythm, intonation style, pronunciation mode, vocabulary selection, and so on. Together, these make up a distinct feature for everyone [1–3].

This work was funded by the National Key Research and Development Program of China under Grant No. 2019YFB1405803.

In the early days, speaker recognition used models such as Dynamic Time Warping (DTW) [4] to calculate the similarity of speakers. Gaussian Mixture Model- Universal Background Models (GMM-UBM) [5,5,6] has improved on this basis. i-vector [7] reduces the computational cost by compressing the speaker vector and can make better use of the channel compensation algorithm. Most speaker recognition requires pre-processing steps. Incorrect pre-processing of recorded speech input can reduce classification performance [8]. Some common methods include noise removal, endpoint detection, pre-emphasis, framing, and normalization [9,10] can be used in pre-processing. After the rise of neural networks, researchers applied DNN [11] directly to speaker classification tasks. This method has the characteristics of large parameters and unclear timing information. The fully connected layer in d-vector was replaced by a feature extraction method similar to one-dimensional convolution in [12–14], which reduces the computational cost. CNNs pays more attention to the speaker information between frames to achieve a better effect. Some networks used artificial features such as MFCC [15,16] and LPCC [17]. However, these features may cause the classifier to miss some speaker-specific information. In order to alleviate this shortcoming, some works used the spectrogram as the input of the network, the others directly used the raw waveform in their network. In order to process the raw waveform, a common choice for researchers is to apply CNNs. Because of the weight sharing feature, the convolution kernel can find a invariant and robust representation. In recent years, there have been many studies on speaker recognition based on the time-domain characteristics of audio using related neural networks [18,19]. Some studies have introduced ResNet [20,21] to deal with more and more complex network structures. However, they are not special in dealing with the first layer of neural network, and are usually the same as other layers in the network. SincNet [22] uses multiple sets of parameterized filters as the first layer of the convolutional network to obtain a personalized solution of the filter. However, SincNet uses the Sinc function as a filter, which only considers limited parameters.

RawNet [23] mixes CNN with LSTM to get better performance. Its input layer is still the same as SincNet, which has the same limitation. This paper alleviates the constraints of the Sinc function on the filter, and enables the filter to directly fit the shape in the frequency domain. This is formed by a set of parameterized linear filtering constraints. This method gives the filter more freedom and enables it to learn more peculiar features.

Our experiments were conducted under challenging but realistic conditions, which are characterized by few data (12–15 s per speaker) and short test sentences (lasting 2 to 6s). The results show that the filter achieves better final performance.

The remainder of the paper is organized as follows. The PF-Net architecture is described in Sect. 2. Section 3 discusses the relation to prior work. The experimental setup and results are outlined in Sect. 4 and Sect. 5 respectively. Finally, Sect. 6 discusses our conclusions.

## 2 The SincNet Architecture

For speech time series signals, the standard first-layer CNN structure is regarded as a time-domain convolution operation [24], which is defined as follows:

$$y[n] = x[n] * h[n] = \sum_{l=0}^{L-1} x[l] \cdot h[n-l] \quad (1)$$

where  $x[n]$  is the speech signal,  $h[n]$  is the filter of length  $L$ , and  $y[n]$  is the output of the filter. In the standard CNN structure, the  $L$  elements of each filter are learned from the data.

In the SincNet structure, the convolution operation uses a predefined function  $g$ , where  $g$  contains only a few learnable variables, defined as follows:

$$y[n] = x[n] * g[n, \theta] \quad (2)$$

SincNet uses the low and high frequencies to filter the original audio in segments. The  $g$  function is defined as a rectangular bandpass filter in the text, and its frequency domain characteristics are as follows:

$$G[f, f_{beg}, f_{end}] = \text{rect}\left(\frac{f}{2f_{end}}\right) - \text{rect}\left(\frac{f}{2f_{beg}}\right) \quad (3)$$

where  $f_{beg}$  and  $f_{end}$  are start cut-off frequency and end cut-off frequency respectively, both of which are learnable.

Unlike SincNet, the filter in PF-Net convolution operation reconstructs the function  $g$  so that  $g$  contains more learnable variables. This is achieved by inserting learnable deformation points in the waveform segments at different low and high frequencies, and divide a frequency domain into multiple one can learn small line segments. The frequency domain characteristics of each segment are expressed as follows:

$$G[f, f_k, f_{k+1}] = \frac{h_{k+1} - h_k}{f_{k+1} - f_k} (f - f_k) + h_k \quad (4)$$

$f_k$  and  $f_{k+1}$  represent the low and high frequencies of each segment,  $h_k$  and  $h_{k+1}$  are their corresponding amplitude intensity in frequency domain.  $f_k$ ,  $f_{k+1}$ ,  $h_k$  and  $h_{k+1}$  are learnable, and the value of  $f$  is within  $[f_k, f_{k+1}]$ .

When  $f$  is not in the interval  $[f_k, f_{k+1}]$ :

$$G[f, f_k, f_{k+1}] = 0, \quad (5)$$

Accumulate each line segment to get the whole frequency band that can be learned. The expression is as follows:

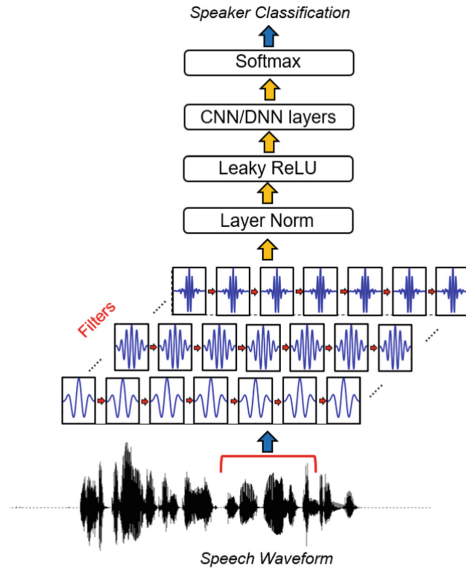
$$G[f, f_{beg}, f_{end}] = \sum_{k=beg}^{end-1} G[n, f_k, f_{k+1}] \quad (6)$$

After inverse Fourier transform [24], the above formula about each line segment is transformed into the time domain expression form as:

$$g[n, f_k, f_{k+1}] = \frac{\Delta(\cos(2\pi f_{k+1}n) - \cos(2\pi f_k n))}{4\pi^2 n^2} - \frac{h_{k+1} \sin(2\pi f_{k+1}n) - h_k \sin(2\pi f_k n)}{2\pi n} \quad (7)$$

where:

$$\Delta = \frac{h_{k+1} - h_k}{f_{k+1} - f_k} \quad (8)$$



**Fig. 1.** Network Architecture of SincNet [22], PF-Net differs from it only in the filter part

The time domain expression of the entire filtering frequency band is:

$$g[f, f_{beg}, f_{end}] = \sum_{k=beg}^{end-1} g[n, f_k, f_{k+1}] \quad (9)$$

Among them, the cutoff frequencies is initialized to a value in the range of  $[0, f_s/2]$ , and  $f_s$  is the sampling rate of the signal. In fact, in order to better extract effective information from low frequencies, both the cutoff frequency and the deformation point frequency are initialized with Mel frequency. To constrain the shape of the filter,  $h_k$  is expressed as:

$$h_k = 1 + \Delta h \quad (10)$$

By constraining the value of  $\Delta h$ , the value of  $h$  can be constrained indirectly. This paper uses  $\Delta h$  instead of  $h$  as the learnable parameters, so that  $h_k$  always learns with 1 as the center. The value of  $\Delta h$  is initialized randomly between  $[-0.1, 0.1]$ . In this way, the deep neural network can learn the shape of each filter. As shown in Fig. 2, the initial shape of the filter in the frequency domain is the same as that of SincNet. After learning, the deformation points set in the frequency band begin to move in different directions by changing the abscissa and ordinate of the points (both of which are learnable parameters). Therefore, PF-Net can obtain richer information than the cut-off frequency.

Finally, in order to smooth the truncation characteristics of the  $g$  function, multiply the  $g$  function by a window function:

$$g_w[n, f_{beg}, f_{end}] = g[n, f_{beg}, f_{end}] \cdot w[n]. \tag{11}$$

Adopt Hamming window [25]:

$$w[n] = 0.54 - -0.46 \cdot \cos\left(\frac{2\pi n}{L}\right). \tag{12}$$

The neural network used in this paper is divided into two parts, the first part is the filter layer, the follow-up is the neural network part composed of CNN/DNN, and finally the posterior probability is output by Softmax. Preprocessing includes removing silence, framing, and maximum normalization operations. Among them, the first layer filter is the key design part, and the network structure is as Fig. 1.

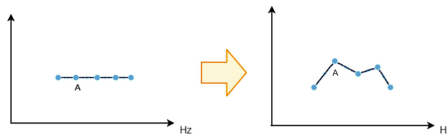


Fig. 2. Learning process of a filter in frequency domain.

### 2.1 Model Properties

- **Flexible parameters:** PF-Net can control the number of parameters in the first convolutional layer. For example, if we consider a convolutional layer with  $F$  channels, each channel uses a filter of length  $L$ , and the number of deformation points is  $S$ , the standard CNN uses  $F \cdot L$  parameters, SincNet uses  $2F$  parameters, and PF-Net uses  $L \cdot F \cdot S$  parameters, if  $F = 80, L = 200, S = 5$ , CNN has 16000 parameters, and SincNet has 160. PF-Net has 800 parameters, somewhere in between. If you change the value of  $S$ , you can change the number of parameters without changing the number of channels. Moreover, if we filter twice the length  $L$ , twice the parameter count of the standard CNN, while the amount of parameters of PF-Net remains unchanged. This provides a very selective filter without actually adding parameters to the optimization

problem. In addition, as the number of parameters increases, the PF-Net network has a more powerful fitting function, and at the same time, it is more difficult to train. The user can freely set the number of deformation points according to the data and actual needs.

- **Interpretability:** Same as SincNet, the features obtained by PF-Net in the first convolutional layer are more interpretable and readable than other methods. The reason for this property of PF-Net is that all deformation points have corresponding positions in frequency domain.

### 3 Related Work

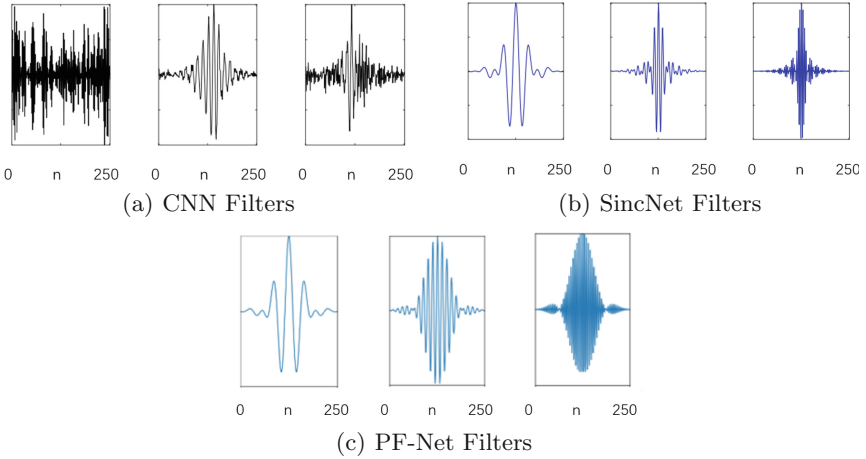
In the process of speaker recognition, feature extraction has attracted much attention. Researchers always assume that the input signal has a stable property in a short enough time interval. Therefore, segmenting the input signal into multiple short-term frames and extracting their feature sequences can better model the audio signal [26]. In the past, people mainly used manual features such as MFCC [15, 16], LPCC [17], and PLP [27]. These methods will inevitably lose part of the speaker information. In the field of CNN, there have been many attempts to process audio using amplitude spectrograms [28–32]. Compared with manual features, the amplitude spectrogram retains more speaker features. However, this method requires careful adjustment of some key hyperparameters such as frame window length, overlap, and type. Therefore, it has become a trend to conduct deep learning directly through raw audio [33–37] has proved its feasibility.

SincNet [22] uses a set of parameterized Sinc filters to directly learn from the original audio. Liu proposed a learnable MFCC [38], but only carried out experiments in speaker verification. In order to obtain the timing characteristics in the original audio, many neural networks [18, 19] used to process timing signals are used for speaker recognition. To deal with more complex neural networks, ResNet [20, 21] was introduced into speaker recognition. These networks pay more attention to the deep structure, but pay less attention to the first layer structure of the network. RawNet [23] mixes CNN with LSTM to get better performance, which has the same first layer as SincNet. However, for each filter, SincNet only has two parameters that can be learned.

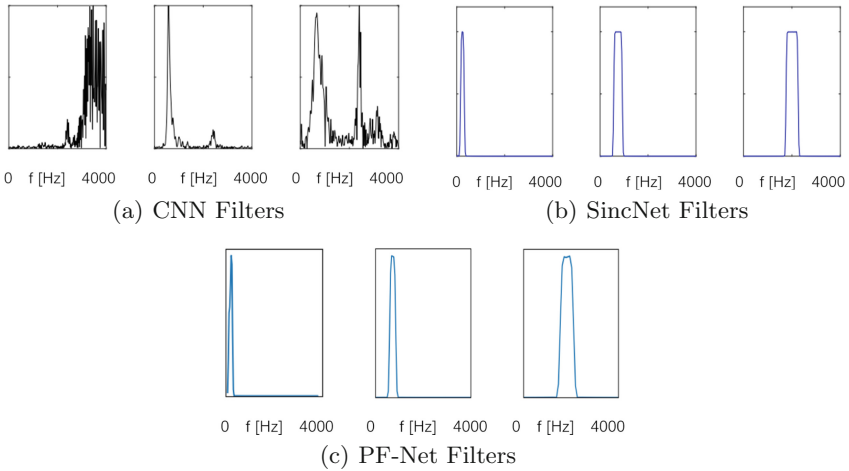
In this paper, a neural network which called PF-Net is proposed to learn the shape of the filter by inserting deformation points into each filter. PF-Net can actively control the number of learnable parameters by controlling the number of deformation points. This kind of filter has a good effect on speaker recognition, especially in the scene with several seconds samples. Each speaker provides a short test statement.

### 4 Experimental Setup

On some public datasets (TIMIT and Librispeech), PF-Net was compared with different baseline systems. In the following sections, we show the detailed settings of the experiment.



**Fig. 3.** Time domain representation of different channel filters earned by neural networks.



**Fig. 4.** Time domain representation of different channel filters earned by neural networks.

### 4.1 Corpora

This paper conducts experiments on the following two datasets.

TIMIT corpora [39], the voice sampling rate of the TIMIT dataset is 16 kHz, which contains a total of 6300 sentences, and 70% of the speakers are male. Most of the speakers are adult whites. Use the TIMIT dataset training set part for the speaker recognition task (same as SincNet), After deleting TIMIT’s calibration sentences (same sentences), each speaker had 8 sentences, of which five sentences were used to train the model, and the remaining three were used for testing.

Librispeech corpora [40], which is a corpus of about 1000 h of 16 kHz reading English speech. It contains 2484 speakers, and the sampling rate is 16 kHz. The training and test speech in the dataset are used in the experiment. After removing the silent part, each sentence lasts for 2–6 s.

## 4.2 PF-Net Setup

Speech signal is divided into frames (frame length 200 ms, overlapping 10 ms) and then input into PF-Net. We insert five deformation points in each filter. As Fig. 2, The filter layer uses 80 filters of length 251, which are calculated by the formula in Sect. 2. Subsequently, the CNN part used two standard convolutional layers, the number of channels was 60, and the size of the convolution kernel was 5. Layer normalization [41] was used for input samples and for all convolutional layers (so is the filter layer). Next, we used 3 fully connected layers containing 2048 neurons, which normalized with batch normalization [42]. The activation function used leaky-ReLU [43]. The transformation point was initialized with mel-scale. We used softmax classifier to output the posterior probability of each frame. On this basis, the results of sentence-level classification were obtained by voting. The RMSprop optimizer was applied, and the learning rate  $lr = 0.001$ ,  $\alpha = 0.95$ ,  $\varepsilon = 10^{-7}$ . The batch size is 128. All hyper-parameters are optimized on TIMIT, as is Librispeech. The speaker verification system can directly take the softmax posterior score corresponding to the claimed identity [44].

## 4.3 Baseline Setups

This paper compares PF-Net with several different systems.

First of all, CNN network with raw waveform as input was applied. Its network structure is the same as PF-Net in this paper, but the first layer convolution was replaced by PF-Net convolution. Secondly, we compared with manual features, this paper uses Kaldi toolkit [45] to calculate 40 FBANK features, which were calculated every 25 ms, with an overlap time of 10 ms, forming a context window of 200 ms. Layer normalization was also used in FBANK network and processed by CNN network. Finally, Sinc function was used to replace the first layer convolution of CNN. Except for the first layer, the network structure and hyperparameters of PF-Net were all the same as SincNet [22]. For speaker verification experiments, the enrollment and test phase is conducted on Librispeech.

## 5 Results

This section reports the experimental validation of the proposed SincNet. First, we perform a comparison between the filters learned by a SincNet and by a standard CNN. We then compare our architecture with other competitive systems on speaker recognition tasks.

## 5.1 Filter Analysis

Figure 3 and Fig. 4 are visual representations of the time-frequency domain of the filters learned by CNN, SincNet and PF-Net in the librispeech dataset, respectively. Through the analysis of them, we can find that PF-Net has the advantages of both original CNN and SincNet. Figure 3 represents the time-domain waveforms of the three filters. Through observation, we can conclude that the filters learned by standard CNN are always noisy. In contrast, PF-Net and SincNet can learn more interpretable filters. In the time domain, SincNet found a narrower waveform while PF-Net found a wider one. In the frequency response shown in Fig. 3, the standard CNN is more difficult to interpret, and the noise is almost distributed in all frequency bands. For SincNet structure, although the filter can learn the corresponding response of a specific frequency band, its model remains unchanged in amplitude intensity. Figure 4(c) shows that PF-Net is different from both. It not only realizes band-pass filtering, but also is not invariable in amplitude intensity.

## 5.2 Speaker Identification

**Table 1.** Classification Error Rates (CER%) of PF-Net on two datasets

Model	TIMIT	LibriSpeech
CNN-FBANK	0.86	1.55
CNN-Raw	1.65	1.00
SINCNET	0.85	0.96
PF-NET	<b>0.72</b>	<b>0.77</b>

Table 1 shows the comparison of Classification Error Rates (CER%). This table shows the comparison of PF-Net with other networks on the LibriSpeech dataset and TIMIT dataset. The network using CNN raw has a lower accuracy rate on the TIMIT dataset and a higher accuracy rate on the LibriSpeech dataset, which indicates the advantage of the original filter, that is, it is more suitable for datasets with large data volume, which is also a feature of the neural network. CNN network using fbank performs well on data sets with small data volume due to the use of manually extracted features. However, it is not as good as CNN on larger datasets. Sincnet and PF-net are both manually designed features and retain learnable parameters. Both of them combine the learnability of neural network and the experience of manual design. Therefore, in two datasets of different sizes, the CER can still be kept low. Among them, because PF-net has designed more parameters that can be learned, it can learn more complex filter banks, which makes PF-net outperform sincnet in datasets. This is because larger datasets need more complex filters for characterization. Due to the small amount of data in TIMIT, Sincnet is not very different from PF-net. However, with the

increase of the amount of data, simple bandwidth learning cannot adapt to more complex data, which makes a greater gap between the two in LibriSpeech.

The LibriSpeech dataset was fast in the first 500 rounds. Although the speed slowed down thereafter, there was still a performance improvement, and the training ended in the 2900th round. The Frame Error Rate (FER%) of LibriSpeech is 0.3, which is better than the result on the TIMIT dataset.

### 5.3 Speaker Verification

Table 2 shows the comparison of Equal Error Rate (EER%). This table shows the performance of the PF-Net on the LibriSpeech dataset. It can be seen that PF-Net performs better than both a CNN trained on standard FBANK coefficients and CNN trained on the raw waveform in speaker verification task. Although the performance of PF-Net is close to SincNet in this task, it still has certain advantages. Through the above experiments, it can be seen that PF-net improves the baseline in speaker identification more than in speaker verification, which may be because the speaker verification task is more prone to over fitting than speaker recognition.

**Table 2.** Equal Error Rates (EER%) of PF-Net on two Librispeech dataset.

Model	LibriSpeech
CNN-FBANK	0.37
CNN-Raw	0.36
SINCNET	0.32
PF-NET	<b>0.30</b>

## 6 Conclusion

In this paper, PF-Net is proposed, which is a neural architecture for directly processing waveform audio. Our model is a change of SincNet, which forms a loose constraint on the shape of the filter through effective parameterization and retains a certain order. Compared with SincNet, PF-Net has more freedom and can adjust the number of parameters freely. PF-Net has carried out extensive evaluation for challenging speaker recognition tasks, showing its performance advantages in speaker recognition. In addition to improving the performance, PF-Net and SincNet have the same advantages, that is, the convergence speed of standard CNN is improved, and the computational efficiency is improved by using the symmetry of the filter. In future work, we plan to evaluate PF-Net performance on datasets in other languages (such as Cn-Celeb) or in more challenging environments (such as VoxCeleb). Although only speaker recognition experiments are carried out in this paper, in theory, PF-Net, a time series processing method, should also work in other fields. Therefore, our future

work will be extended to emotion recognition and singer recognition. We have ever uploaded the earlier version of this paper in arXiv, and now we have the opportunity to publish it in MobileMedia. We also open related codes in Github ([github.com/TAN-OpenLab/PF-NET](https://github.com/TAN-OpenLab/PF-NET)).

## References

1. Atal, B.: Automatic recognition of speakers from their voices. *Proc. IEEE* **64**(4), 460–475 (1976). <https://doi.org/10.1109/PROC.1976.10155>
2. Shriberg, E., Ferrer, L., Kajarekar, S., Venkataraman, A., Stolcke, A.: Modeling prosodic feature sequences for speaker recognition. *Speech Commun.* **46**(3), 455–472 (2005). <https://doi.org/10.1016/j.specom.2005.02.018>. Quantitative Prosody Modelling for Natural Speech Description and Generation
3. Furui, S.: An overview of speaker recognition technology. In: Lee, C.H., Soong, F.K., Paliwal, K.K. (eds.) *Automatic Speech and Speaker Recognition*. The Kluwer International Series in Engineering and Computer Science, vol. 355, pp. 31–56. Springer, Boston (1996). [https://doi.org/10.1007/978-1-4613-1367-0\\_2](https://doi.org/10.1007/978-1-4613-1367-0_2)
4. Rabiner, L., Rosenberg, A., Levinson, S.: Considerations in dynamic time warping algorithms for discrete word recognition. *IEEE Trans. Acoust. Speech Signal Process.* **26**(6), 575–582 (1978). <https://doi.org/10.1109/TASSP.1978.1163164>
5. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. *Digit. Signal Process.* **10**(1), 19–41 (2000). <https://doi.org/10.1006/dspr.1999.0361>
6. Pathak, M.A., Raj, B.: Privacy-preserving speaker verification and identification using Gaussian mixture models. *IEEE Trans. Audio Speech Lang. Process.* **21**(2), 397–406 (2013). <https://doi.org/10.1109/TASL.2012.2215602>
7. Liu, G., Hansen, J.H.L.: An investigation into back-end advancements for speaker recognition in multi-session and noisy enrollment scenarios. *IEEE/ACM Trans. Audio Speech Lang. Process.* **22**(12), 1978–1992 (2014). <https://doi.org/10.1109/TASLP.2014.2352154>
8. Singh, N.: A critical review on automatic speaker recognition. *Sci. J. Circ. Syst. Signal Process.* **4**, 14 (2015). <https://doi.org/10.11648/j.cssp.20150402.12>
9. Yilmaz, E., McLaren, M., van den Heuvel, H., van Leeuwen, D.A.: Language diarization for semi-supervised bilingual acoustic model training. In: 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 91–96 (2017). <https://doi.org/10.1109/ASRU.2017.8268921>
10. Cutajar, M., Gatt, E., Grech, I., Casha, O., Micallef, J.: Comparative study of automatic speech recognition techniques. *IET Signal Proc.* **7**(1), 25–46 (2013). <https://doi.org/10.1049/iet-spr.2012.0151>
11. Variani, E., Lei, X., McDermott, E., Moreno, I.L., Gonzalez-Dominguez, J.: Deep neural networks for small footprint text-dependent speaker verification. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4052–4056 (2014). <https://doi.org/10.1109/ICASSP.2014.6854363>
12. Poddar, A., Sahidullah, M., Saha, G.: Speaker verification with short utterances: a review of challenges, trends and opportunities. *IET Biometr.* **7** (2017). <https://doi.org/10.1049/iet-bmt.2017.0065>
13. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. *IEEE Trans. Audio Speech Lang. Process.* **19**(4), 788–798 (2011). <https://doi.org/10.1109/TASL.2010.2064307>

14. Debnath, S., Soni, B., Baruah, U., Sah, D.K.: Text-dependent speaker verification system: a review. In: 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), pp. 1–7 (2015). <https://doi.org/10.1109/ISCO.2015.7282386>
15. Balpande, M., Sansare, R., Padelkar, T., Shinde, V.: Speaker recognition based on mel-frequency cepstral coefficients and vector quantization. In: 2021 IEEE Bombay Section Signature Conference (IBSSC), pp. 1–6 (2021). <https://doi.org/10.1109/IBSSC53889.2021.9673167>
16. Zhang, N., Yao, Y.: Speaker recognition based on dynamic time warping and gaussian mixture model. In: 2020 39th Chinese Control Conference (CCC), pp. 1174–1177 (2020). <https://doi.org/10.23919/CCC50068.2020.9188632>
17. Rodriguez-Porcheron, D., Faundez-Zanuy, M.: Speaker recognition with a MLP classifier and LPCC codebook. In: Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP99 (Cat. No.99CH36258). vol. 2, pp. 1005–1008 (1999). <https://doi.org/10.1109/ICASSP.1999.759872>
18. Zhao, Y., Zhou, T., Chen, Z., Wu, J.: Improving deep CNN networks with long temporal context for text-independent speaker verification. In: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6834–6838 (2020). <https://doi.org/10.1109/ICASSP40776.2020.9053767>
19. Utomo, Y.F., Djamal, E.C., Nugraha, F., Renaldi, F.: Spoken word and speaker recognition using MFCC and multiple recurrent neural networks. In: 2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI), pp. 192–197 (2020). <https://doi.org/10.23919/EECSI50503.2020.9251870>
20. Jakubec, M., Lieskovska, E., Jarina, R.: Speaker recognition with ResNet and VGG networks. In: 2021 31st International Conference Radioelektronika (RADIOELEKTRONIKA), pp. 1–5 (2021). <https://doi.org/10.1109/RADIOELEKTRONIKA52220.2021.9420202>
21. Zhou, T., Zhao, Y., Wu, J.: ResNeXt and Res2Net structures for speaker verification. In: 2021 IEEE Spoken Language Technology Workshop (SLT), pp. 301–307 (2021). <https://doi.org/10.1109/SLT48900.2021.9383531>
22. Ravanelli, M., Bengio, Y.: Speaker recognition from raw waveform with SincNet. In: 2018 IEEE Spoken Language Technology Workshop (SLT), pp. 1021–1028 (2018). <https://doi.org/10.1109/SLT.2018.8639585>
23. Weon Jung, J., Heo, H.S., ho Kim, J., Jin Shim, H., Yu, H.J.: RawNet: advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification (2019)
24. Rabiner, L., Schafer, R.: Theory and Applications of Digital Speech Processing, 1st edn. Prentice Hall Press, USA (2010)
25. Mitra, S.K.: Digital Signal Processing. McGraw Hill (2006)
26. Malik, S., Afsar, F.A.: Wavelet transform based automatic speaker recognition. In: 2009 IEEE 13th International Multitopic Conference, pp. 1–4 (2009). <https://doi.org/10.1109/INMIC.2009.5383083>
27. Li, Q., Huang, Y.: An auditory-based feature extraction algorithm for robust speaker identification under mismatched conditions. IEEE Trans. Audio Speech Lang. Process. **19**(6), 1791–1801 (2011). <https://doi.org/10.1109/TASL.2010.2101594>

28. Zhang, C., Koishida, K., Hansen, J.H.L.: Text-independent speaker verification based on triplet convolutional neural network embeddings. *IEEE/ACM Trans. Audio Speech Lang. Process.* **26**(9), 1633–1644 (2018). <https://doi.org/10.1109/TASLP.2018.2831456>
29. Kadyrov, S., Turan, C., Amirzhanov, A., Ozdemir, C.: Speaker recognition from spectrogram images. In: 2021 IEEE International Conference on Smart Information Systems and Technologies (SIST), pp. 1–4 (2021). <https://doi.org/10.1109/SIST50301.2021.9465954>
30. Nehra, N., Sangwan, P., Kumar, D.: Speaker identification system using CNN approach. In: 2021 International Conference on Industrial Electronics Research and Applications (ICIARA), pp. 1–6 (2021). <https://doi.org/10.1109/ICIARA53202.2021.9726767>
31. Dinkel, H., Chen, N., Qian, Y., Yu, K.: End-to-end spoofing detection with raw waveform CLDNNs. In: Proceedings of ICASSP, pp. 4860–4864 (2017)
32. Seki, H., Yamamoto, K., Nakagawa, S.: A deep neural network integrated with FilterBank learning for speech recognition. In: Proceedings of ICASSP, pp. 5480–5484 (2017)
33. Palaz, D., Magimai-Doss, M., Collobert, R.: Analysis of CNN-based speech recognition system using raw speech as input. In: Proceedings of Interspeech (2015)
34. Sainath, T.N., Weiss, R.J., Senior, A.W., Wilson, K.W., Vinyals, O.: Learning the speech front-end with raw waveform CLDNNs. In: Proceedings of Interspeech (2015)
35. Hoshen, Y., Weiss, R., Wilson, K.W.: Speech acoustic modeling from raw multi-channel waveforms. In: Proceedings of ICASSP (2015)
36. Sainath, T.N., Weiss, R.J., Wilson, K.W., Narayanan, A., Bacchiani, M., Senior, A.: Speaker localization and microphone spacing invariant acoustic modeling from raw multichannel waveforms. In: Proceedings of ASRU (2015)
37. Tüske, Z., Golik, P., Schlüter, R., Ney, H.: Acoustic modeling with deep neural networks using raw time signal for LVCSR. In: Proceedings of Interspeech (2014)
38. Liu, X., Sahidullah, M., Kinnunen, T.: Learnable MFCCS for speaker verification. In: 2021 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5 (2021). <https://doi.org/10.1109/ISCAS51556.2021.9401593>
39. Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., Dahlgren, N.L.: DARPA TIMIT acoustic phonetic continuous speech corpus CDROM (1993)
40. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech: an ASR corpus based on public domain audio books. In: Proceedings of ICASSP, pp. 5206–5210 (2015)
41. Ba, J., Kiros, R., Hinton, G.E.: Layer normalization. *CoRR* abs/1607.06450 (2016)
42. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of ICML, pp. 448–456 (2015)
43. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of ICML (2013)
44. Muckenhirn, H., Magimai-Doss, M., Marcell, S.: Towards directly modeling raw speech signal for speaker verification using CNNs. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4884–4888 (2018). <https://doi.org/10.1109/ICASSP.2018.8462165>
45. Povey, D., et al.: The Kaldi speech recognition toolkit. In: Proceedings of ASRU (2011)