Complex & Intelligent Systems (2021) 7:1749–1757
https://doi.org/10.1007/s40747-020-00172-1

ORIGINAL ARTICLE

Speaker recognition based on characteristic spectrograms and an improved self-organizing feature map neural network

Yanjie Jia[1] · Xi Chen[2] · Jieqiong Yu[2] · Lianming Wang[1,2] · Yuanzhe Xu[1] · Shaojin Liu[3] · Yonghui Wang[4]

Received: 16 March 2020 / Accepted: 18 June 2020 / Published online: 29 June 2020

Abstract

To obtain a speaker's pronunciation characteristics, a method is proposed based on an idea from bionics, which uses spectrogram statistics to achieve a characteristic spectrogram to give a stable representation of the speaker's pronunciation from a linear superposition of short-time spectrograms. To deal with the issue of slow network training and recognition speed for speaker recognition systems on resource-constrained devices, based on a traditional SOM neural network, an adaptive clustering self-organizing feature map SOM (AC-SOM) algorithm is proposed. This algorithm automatically adjusts the number of neurons in the competition layer based on the number of speakers to be recognized until the number of clusters matches the number of speakers. A 100-speaker database of characteristic spectrogram samples was built and applied to the proposed AC-SOM model, yielding a maximum training time of only 304 s, with a maximum sample recognition time of less than 28 ms. Comparing to other approaches, the proposed method offers greatly improved training and recognition speed without sacrificing too much recognition accuracy. The promising results suggest that the proposed method satisfies real-time data processing and execution requirements for edge intelligence systems better than other speaker recognition methods.

Keywords Speaker recognition · Characteristic spectrogram · Adaptive clustering · Neural network · Deep learning · Edge intelligence

Introduction

Speaker recognition, also known as voiceprint recognition, is an important branch of speech signal processing. It is a biometric identification technology that automatically detects a given speaker by extracting parameters representing his or her speech characteristics via a computer [1, 2]. Human speech is generated by the combined action of several organs, i.e. the lungs, vocal tract, vocal cords, and lips. Because of this complex structure, we can obtain features expressing human pronunciation characteristics by statistically analyzing the information carried by speech signals [3]. These features can be broadly categorized into five common types: short-term spectral features, voice source features, spectro-temporal features, prosodic features, and high-level features. Many speaker recognition systems use several of these features in parallel, including different speech aspects and employing them in complementary ways to achieve more accurate recognition [4].

Nowadays, the speech feature extraction techniques commonly used in speaker recognition systems include Linear Prediction Cepstral Coefficients (LPCC), Mel Frequency Cepstral Coefficients (MFCC) [5], Perceptual Linear Predictive analysis [6], first- and second-order differential coefficients of cepstrum [7], and Relative Spectral Analysis (RASTA) filters [8]. Unlike one-dimensional characteristic parameters, spectrogram [9] is a more intuitive, compact, and efficient representation that carries acoustic feature information in the form of a two-dimensional pattern and includes rich acoustic features such as the energy, pitch, fundamental frequency, and formant. These features are valuable for automated speech and speaker recognition systems and are widely used tools for speech analysis. Many

✉ Lianming Wang
lm.wang@protonmail.com

1 School of Ocean Science and Technology, Hainan Tropical Ocean University, Sanya, China

2 Institute of Computational Intelligence, Northeast Normal University, Changchun, China

3 School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA

4 Department of Computer Science, Prairie View A&M University, Prairie View, TX, USA

researchers used spectrogram as the acoustic feature combining with artificial neural network method for speaker recognition. Li et al. [10] proposed a speaker recognition method that used spectrograms as speech signal features and learning vector quantization (LVQ) neural networks as feature classifiers. Liu et al. [11] performed speaker recognition based on the spectrogram and CNN neural network and the recognition method has good recognition ability in terms of accuracy.

Common speaker recognition methods include hidden Markov models (HMMs) [12], Gaussian mixture models (GMMs) [13], vector quantization [14], dynamic time warping, support vector machines (SVMs) [15], and artificial neural networks. For more than two decades, Gaussian Mixture Model-Universal Background Model (GMM-UBM) has become a widely used paradigm in speaker recognition systems because of its good performance in speaker recognition [16]. However, UBM is used to represent the entire speaker acoustic characteristics of the space. It is required to provide a large amount of background speaker speech data, which increases the time of model training and the complexity of subsequent testing. In recent years, the application of deep learning technology in the speaker recognition field [17, 18] has greatly improved both the recognition rate and robustness, and the results obtained with deep learning neural networks continue to encourage the use of neural networks for speaker recognition. But, building deep learning models is highly computationally intensive: training such models requires a large amount of data and a high-performance central processing unit (CPU) to run the back-propagation algorithm. In more resource-constrained devices, some practical edge intelligence [19] applications must focus more on energy consumption and efficiency when processing and aggregating data to meet their real-time data processing and execution requirements. As the number of speakers to be identified increases, the training and recognition speed required for the speaker model increase dramatically, making real-time implementation on such devices more difficult and less practical. An alternative approach must therefore be found which can train the model and identify speakers more efficiently.

SOM neural network can automatically classify the input data without the need of extensive data samples or manual intervention [20]. It can be used to cluster speaker characteristic parameters, which can better reflect the validity and feasibility of speaker pronunciation feature extraction method. Moreover, the network structure is simple and supports real-time processing, which can satisfy the need for rapid training and recognition of speaker recognition applications. However, the number and structure of the neurons in the competitive layer will affect the classification performance.

This paper introduces a self-built database with diverse sample content, wide regional speaker accent, and considerable speaker age span. The database can be used to achieve the speaker recognition target that is not related to the text. Based on an idea from bionics, this paper proposes a speaker recognition method, which first uses short-time spectrogram statistics to obtain a characteristic spectrogram representing the speaker's pronunciation characteristics (a visual representation of the speaker's stable pronunciation features), and then uses an adaptive clustering self-organizing feature map (AC-SOM) neural network for adaptive cluster learning. This method greatly improves the system's training and recognition speeds without significantly impacting the recognition rate.

For the rest of the paper, the proposed method is described in details in Section "Proposed method", followed by the demonstration of the performance of the proposed method in Section "Experiments". Finally, some concluding remarks and discussions are given in Section "Summary and conclusion".

Proposed method

Characteristic spectrograms

Creating short-time spectrograms

Spectrograms are speech spectrum maps, originally devised during World War II to detect submarines and decipher enemy codes, but later were used in the linguistics field [21]. The vertical and horizontal axes of a gray scale spectrogram represent frequency and time, respectively, whereas each pixel's gray scale value reflects the signal's energy density at the corresponding time and frequency, as shown in Fig. 1. Such spectrograms can show the variations in the

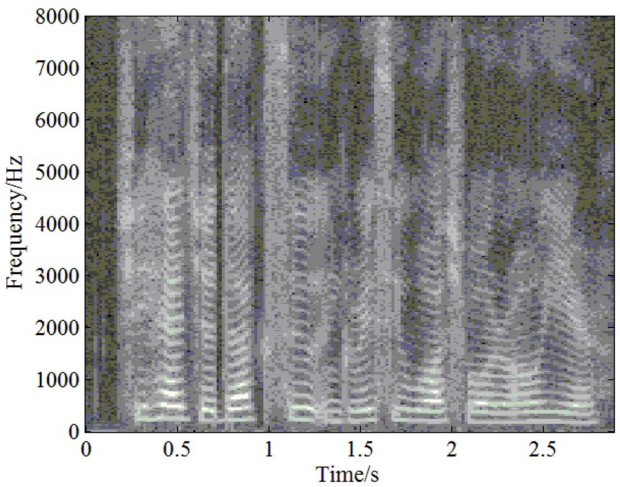


Fig. 1 Grayscale spectrogram

fundamental frequency, pitch period, and formant intensity in the speaker's utterance over time in a two-dimensional way. They are generally used to represent the long-term frequency characteristics of a speech signal, but cannot reflect detailed pronunciation characteristics.

From a physiological perspective, given the short-term stationary characteristics of human pronunciation, long utterances can be divided into several shorter speech segments, each represented as one frame. Each segment of the short-term speech signal can then be regarded as a short-time stationary signal. A short-time spectrogram, as shown in Fig. 2, can be obtained by calculating the signal's power spectral density (PSD) of each frame.

Creating characteristic spectrograms

Humans can recognize other people through their voices with the recognition of their vocal characteristics. This process can be abstracted as conducting a statistical analysis on the pronunciation characteristics. By calculating and storing such statistics, speaker's main pronunciation characteristics can be obtained, allowing for text-independent speaker recognition. Based on this idea, pronunciation characteristics can be extracted by linearly superimposing several short-time spectrograms of a given speaker at the same frequency for a certain time period.

Let C represent the value of a pixel in the superimposed spectrogram. This is calculated from C_i, the corresponding pixel value in the i-th original spectrogram, and the number of superimposed spectrograms N as follows:

$$C = \sum_{i=1}^{N} \frac{255 - C_i}{N}. \tag{1}$$

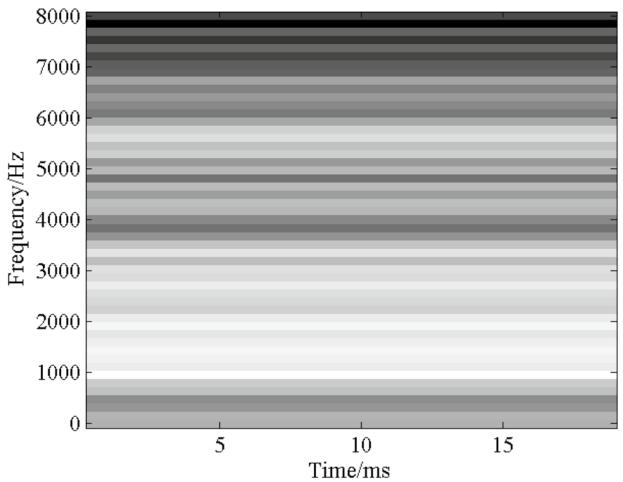


Fig. 2 Short-time spectrogram

The pixel values C lie between 0 and 255, and is represented as gray scale values. The gray scale values in the superimposed image represent the energy distribution of the speaker's utterance over different frequencies during the given time period. When calculating long-time statistics, the energy distribution will tend to be stable, as will the result superimposed spectrogram, making it suitable for representing the speaker's pronunciation characteristics. To reduce the number of samples and obtain more stable pronunciation features, this paper uses two linear superpositions to obtain the speaker's characteristic spectrograms in actual experiment.

Figure 3 gives an overview of the characteristic spectrogram construction process. Superimposing multiple short-time spectrograms may cause the energy to extend beyond the gray scale range; therefore, each short-time spectrogram should first be normalized. We also apply pre-emphasis equivalent to a 6 dB/octave high-frequency lifting filter. Because of glottis excitation and the effect of nose and mouth radiation, the high-frequency end of the average speech frequency spectrum drops by about 6 dB/octave above 800 Hz. The pre-emphasis is used to strengthen the signal's high-frequency formant and smooth the short-time spectrum, thereby to eliminate DC drift, to suppress random noise, and to improve the energy of the spectrum's light and high parts.

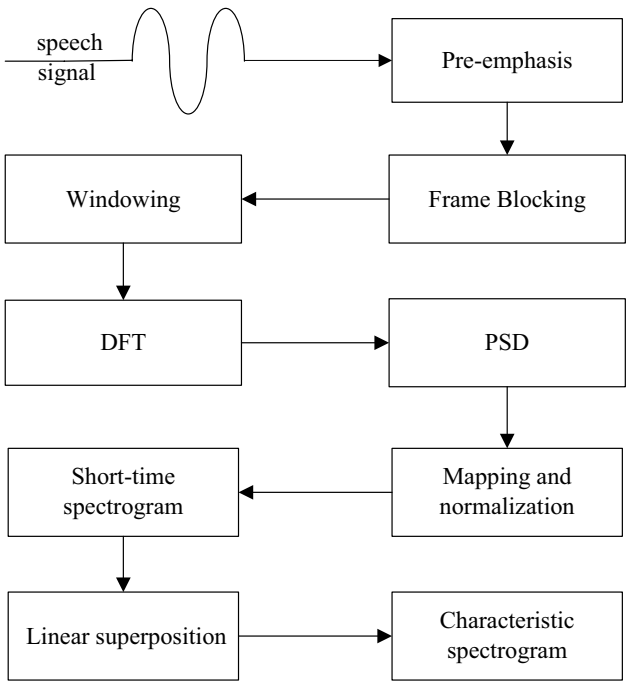


Fig. 3 Block diagram of the proposed process for constructing the speaker's characteristic spectrogram

AC-SOM neural networks

A SOM network receives external input and divides it into different regions according to the input data distribution. Each region responds differently to the input data via this automatic process. It can represent the input space of the training samples discretely and reduce the data dimensionality [22, 23].

Network structure

SOM networks are two-layered feed forward fully connected networks. The first layer is the input layer, which transfers external information to the output layer, also known as the competitive layer. The number of competitive layer neurons depends on the specific task, and each neuron is connected to all the input layer neurons and has its own weight vector. We can think of the output layer as being similar to the cerebral cortex in certain ways, whereas the input layer is similar to the organism's senses, and the connection weights act as information transmission channels.

Traditionally, the output layer of a SOM network forms a one- or two-dimensional array, as shown in Fig. 4a, and the number and arrangement of neurons must be defined in advance. Therefore, for fixed-dimension input data, several attempts are needed to determine the number of competitive layer neurons. To select a suitable number of competitive layer neurons for our clustering problem, we propose to instead use an adaptive clustering SOM (AC-SOM) network structure, as shown in Fig. 4b, which can automatically adjust and increase the number of competitive layer neurons as the number of speakers to be recognized increases.

Adaptive learning and clustering algorithm

We construct a SOM network with n input layer neurons and $m \times m$ competitive layer neurons. The connection weight between each input layer neuron and the output layer neuron with coordinates (i,j). $i,j \in [1,m]$ is given by $\mathbf{W}_{ij} = (w_{ij1}\ w_{ij2} \ldots w_{ijn})^{\mathrm{T}}$. If there are S speakers, with q samples in each characteristic spectrogram, then the total number of samples is $M = S \times q$. The speaker recognition and clustering process for the AC-SOM algorithm is as follows.

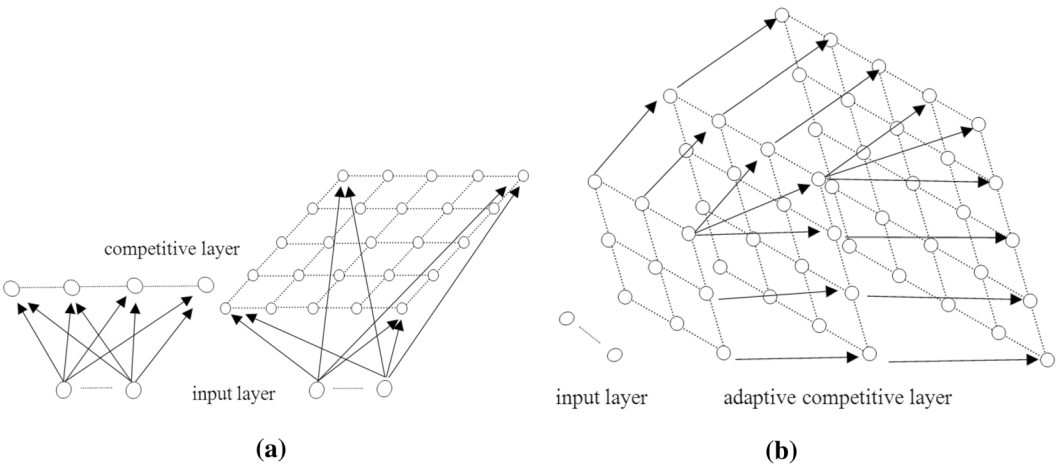
1. *Initialization*: Set the number of competitive layer neurons to $m \times m$, the connection weight to $\mathbf{W}$, the winning neighborhood radius to r_0, the learning rate to α_0, the maximum number of iterations to λ, and the current iteration to $t, t \in [1, \lambda]$.
2. *Sampling*: Take one sample $\mathbf{X}_p = (x_1\ x_2 \ldots x_{\mathrm{n}})^{\mathrm{T}}, p \in [1, \mathrm{M}]$ from the M samples as the network input.
3. *Similarity calculation*: Calculate the Euclidean distances of all the neurons from the input vector $\mathbf{X}_p$ to the output layer, and find the minimum distance. This calculation can be expressed as

$$D(\mathbf{X}_p) = \arg\min_{i,j} \left\| \mathbf{X}_p - \mathbf{W}_{i,j} \right\|. \tag{2}$$

We call the output layer neuron with the minimum distance the "winning neuron," with coordinates (k_1, k_2).

4. *Updating*: Adjust the weights of all neurons in the neighborhood of the winning neuron as follows:

$$h(\rho, t) = \begin{cases} 1 & \rho \le r(t) \\ 0 & \rho > r(t) \end{cases} \quad \rho = \sqrt{(k_1 - i)^2 + (k_2 - j)^2}, \tag{3}$$



(a) (b)

Fig. 4 Structures of **a** traditional SOM and **b** AC-SOM adaptive networks

$$\mathbf{W}_{ij}(t+1) = \mathbf{W}_{ij}(t) + h(\rho, t) \cdot \alpha(t) \cdot (\mathbf{X}_p - \mathbf{W}_{ij}(t)), \quad (4)$$

where $h(\rho, t)$ is the neighborhood function, ρ is the distance between the other output layer neurons and the winning neuron, $r(t)$ is the winning neighborhood's radius (the neurons with $\rho \leq r(t)$ representing the winning neuron's neighborhood), and $a(t)$ is the learning rate. The learning rate decreases at later iterations to ensure the learning process eventually converges.

5. If $p < \mathrm{M}$, increment p and repeat Steps 2–5, otherwise go to Step 6.
6. Adjust the learning rate and winning neighborhood radius as following:

$$\alpha(t) = \alpha_0\left(1 - \frac{t}{\lambda}\right) \quad (5)$$

$$r(t) = r_0 \cdot \left(1 - \frac{t}{\lambda}\right). \quad (6)$$

7. As long as the learning rate is greater than a given minimum value and this is not the last iteration (i.e., $t < \lambda$), increment t and go to Step 2, otherwise go to Step 8.
8. *Counting the number of clustering centers:* First, count the number of times each neuron was the winning neuron over the q training samples for each of the S speakers. The most frequently winning neuron for each speaker represents that speaker's category, which is the speaker's clustering center. Finally, count the number of distinct clustering centers.
9. If the number of clustering centers is strictly less than the number of speakers S, increase the maximum number of iterations or increment the number of output layer neurons m and go to Step 1; otherwise terminate.

With this adaptive learning and clustering algorithm, the number of competitive layer neurons can be automatically adjusted and increased as the number of speakers to be recognized increases, which reduces the time of many attempts to select the suitable number of competitive layer neurons.

Experiments

Database construction and experimental environment

For the experiments, we created a Chinese language database containing recordings of 100 speakers (50 men and 50 women). Each recording was approximately 7 min in length and was created in a laboratory using PC audio recording software at a sampling frequency of 16 kHz and saved in WAV format. Each speaker received a different script from a novel, and spoke at a normal rate.

Using these speech samples, we then created a database of the speakers' characteristic spectrograms. Each speaker recording was intercepted and broken down into 4000 short-time spectrograms. Then, as shown in Fig. 5, one-superposition spectrograms were created from each group of 40 short-time spectrograms, yielding 100 such spectrograms per speaker. In image processing, linear superimposition refers to performing weighted average operation on corresponding pixels of multiple images. In these experiments, 40 short-time spectrograms are used as a group to carry out the weighted average operation with a weight of "1", which we call one-superposition. After that, the same weighted average operation is carried out again in a group of 10 one-superposition spectrograms, which we call quadratic-superposition. To reduce the number of samples and obtain more stable pronunciation features, groups of 10 one-superposition spectrograms were combined using quadratic linear superposition to eventually obtain 10 characteristic spectrograms per speaker. Thus, we generated 1000 characteristic spectrogram samples from our 100-speaker database.

These experiments were conducted using MATLAB R2010a, running on a PC with an Intel Core i7-4790 CPU with an NVIDIA GeForce GT 740 GPU and 8 GB RAM, on the Windows 7 64-bit operating system.

Speaker recognition experiment

Figure 6 provides an overview of the speaker recognition system. In this experiment, we used 80% of each speaker's data for training and the remaining 20% for testing, with none of the samples in the test set being used for training. Based on the cubic convolution method, the characteristic spectrograms with an output size of 420×560 was downsampled to sample pictures with a size of 42×56, and then the sample pictures were sequentially input into the AC-SOM neural network for training, which had an input layer dimension of $n = 42 \times 56 = 2352$. After training on S

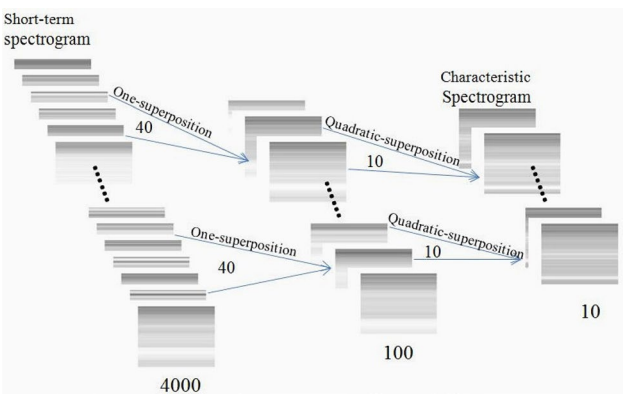


Fig. 5 Generated characteristic spectrograms for a single speaker

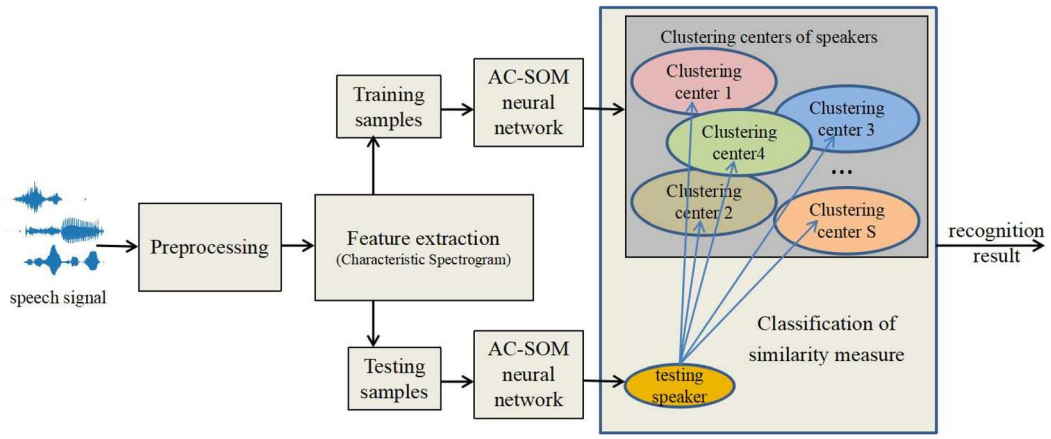


Fig. 6 Overview of the experimental speaker identification system

speakers, we finally obtained S clustering centers. We then classified the test samples by calculating the Euclidean distance similarity between each sample and all the clustering centers, defining the recognition result for each sample as the clustering center with the highest similarity.

Results

In this section, the proposed method is evaluated by performing various speaker recognition experiments using the database described above. The performance is compared to those of the other speaker recognition methods, i.e., deep belief net (DBN) [24], convolutional neural network (CNN) [25], and back-propagation (BP) network [26]. In addition, our approach is also compared to other feature extraction methods, namely MFCC and LPCC. For training speed comparison, all the methods use GPU. In all experiments, we calculated the recognition rate as the number of correct matches out of the total number of tested samples, as follows:

$$\text{Recognition rate} = \frac{\text{Number of correct matches}}{\text{Total number of speakers}} \times 100\%.$$

With this formula, the test and training set recognition rates were obtained by testing all the test and training set samples, respectively. The training time is defined as the time (in seconds) required for the speaker recognition system to complete the training process, whereas the total test time is the time taken to classify all test set samples and the single test time is the average time taken to classify one test sample.

Effect of the number of speakers on the recognition rate, training speed, and recognition speed

To investigate the effect of the number of speakers on the average training and test set recognition rates, we tested the system with 20, 30, 50, 70, 90, and 100 speakers under the same experimental conditions. We ran each experiment three times, and the average results are given in Table 1.

From Table 1, we can see that the training set recognition rate is typically slightly higher than that of the test set, and the recognition is higher for smaller numbers of speakers. As the number of speakers increases, the recognition rate drops.

Regarding the time taken, Table 1 also shows the average training, total test, and single test times. This indicates that, although the network's training and recognition times increased as the number of speakers increased, even with 100 speakers, the training time was still only about 5 min, and the single test time was less than 28 ms, showing that both training and recognition were still remarkably fast.

Table 1 Effect of the number of speakers on the recognition rate, training speed, and recognition speed

No. of speakers	Recognition rate (training set)(%)	Recognition rate (test set) (%)	Training time (s)	Total test time (s)	Single test time (ms)
20	99.4	100.0	12.1	0.2	4.1
30	98.3	96.7	18.3	0.5	7.9
50	92.5	91.0	70.2	1.2	12.4
70	91.6	87.9	139.6	2.5	18.1
90	86.4	83.9	201.4	4.4	24.3
100	86.4	82.5	304.8	5.6	27.8

Feature extraction method comparison

To further evaluate the performance of the proposed spectrogram statistics-based feature extraction method, it was compared to two other feature extraction methods under the same experimental conditions with the same data, for 30 and 50 speakers. After extracting the features, they were imported into the AC-SOM network for training and, later, recognition. The same number of speech signals was used to extract the MFCC and LPCC parameters and for generating short-time spectrograms. In this experiment, the feature extraction order for the LPCC and MPCC techniques was 13. As before, 80% of each speaker's data was used for training and the remaining 20% for testing.

These three methods were used to extract characteristic parameters for each speaker, which were then used to train the speaker recognition system. Figure 7 and Table 2 show the results, averaged over three runs of the experiment.

As shown in Fig. 7 and Table 2, under the same recognition method, the LPCC speaker feature extraction method has the lowest recognition rate. The MFCC feature extraction method recognition rate is higher than that of LPCC. But both recognition rates are less than 90%. The proposed spectrogram-based feature extraction method performs better than the commonly used MFCC and LPCC methods for the same number of speakers, and can effectively extract speaker pronunciation features.

Table 2 Speaker recognition performance comparison for three different feature extraction methods

No. of speakers	Feature extraction methods	Recognition rate (training set) (%)	Recognition rate (test set) (%)
30	MFCC	89.7	88.4
	LPCC	85.9	84.1
	Characteristic spectrogram	98.3	96.7
50	MFCC	88.3	85.1
	LPCC	80.7	79.2
	Characteristic spectrogram	92.5	91.0

Effect of different recognition methods on the recognition rate and speed

Finally, we compared our approach to three other speaker recognition methods (DBN, CNN, and BP network) in terms of recognition rate and speed, for 30, 50, and 100 speakers. We implemented the DBN and CNN using the MATLAB Deep Learning Toolbox. CNN consisted of two convolutional layers, two pooling layers, two fully connected layers, and one softmax layer, whereas the DBN's structure was $2352 \times 1000 \times 500 \times 250 \times 100 \times$ Number of speakers. The average recognition rates and speed over three runs of the experiment are shown in Fig. 8 and Table 3.

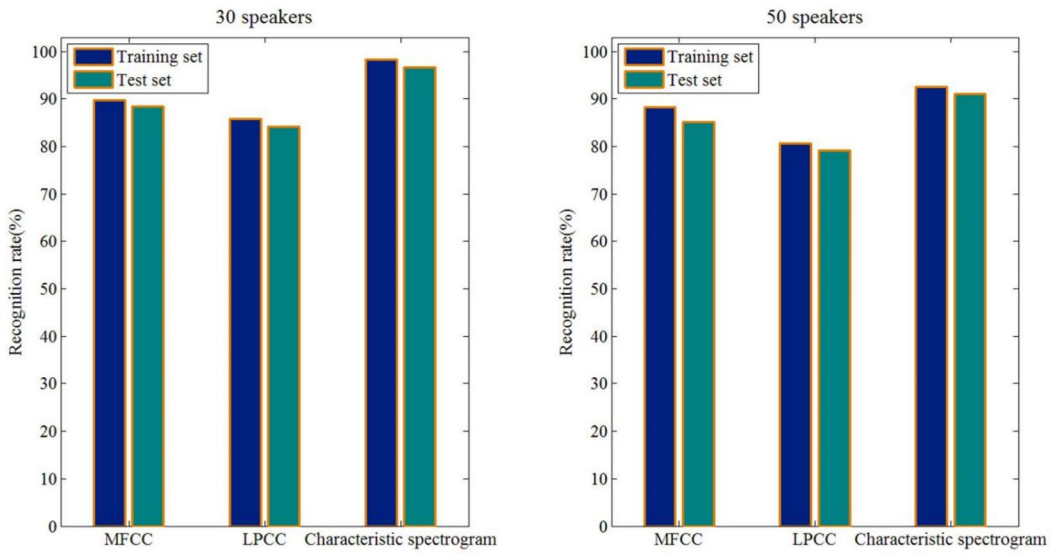


Fig. 7 Speaker recognition performance comparison for three different feature extraction methods

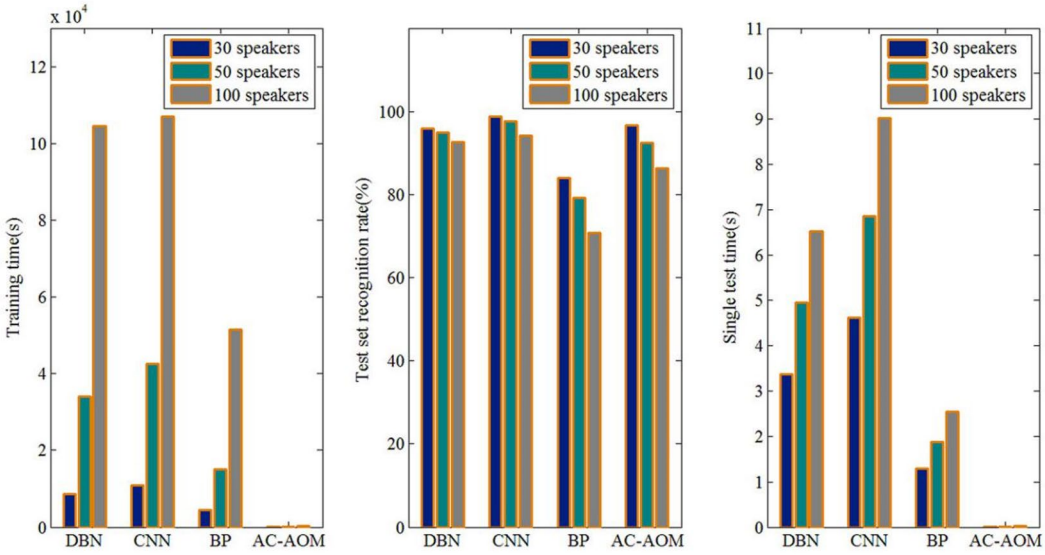


Fig. 8 Recognition rates and speed for four different recognition methods

Table 3 Recognition rates and speed for four different recognition methods

No. of speakers	Recognition methods	Training time (s)	Recognition rate (test set) (%)	Single test time (s)
30	DBN	8600.2	95.8	3.373
	CNN	10,856.1	98.8	4.625
	BP	4530.2	84.0	1.291
	AC-SOM	18.3	96.7	0.008
50	DBN	33,978.7	95.0	4.948
	CNN	42,571.2	97.6	6.854
	BP	15,013.9	79.1	1.872
	AC-SOM	70.181	92.7	0.012
100	DBN	104,574.7	92.5	6.527
	CNN	107,002.5	94.1	9.027
	BP	51,398.0	70.8	2.539
	AC-SOM	304.8	86.4	0.027

As can be seen in Fig. 8 and Table 3, under the same experimental conditions, the CNN recognition rate is the highest, but it sacrifices training speed and recognition speed. The recognition rate of the AC-SOM neural network method proposed in this study is only slightly lower than that of the CNN method, but the training speed and recognition speed of the proposed network are significantly faster than those of the other methods, which is obviously superior to other methods and can meet the needs of real-time applications.

Summary and conclusion

To solve the problems of slow network training speed, low recognition efficiency, and poor application performance on resource-constrained devices, this paper proposes a method which uses short-time spectrogram statistics to obtain stable pronunciation features for each speaker, and then recognizes speakers with an AC-SOM neural network-based adaptive clustering method. A Chinese language database was created, which contains recordings of 100 speakers. An

effective feature extraction method was proposed to obtain the speakers' characteristic spectrograms. Then the characteristic spectrograms were used to test the recognition effectiveness and speed of the proposed AC-SOM model. The experimental results show that a speaker's characteristic spectrogram can reflect not only his or her pronunciation details but also stable pronunciation characteristics, thereby can effectively characterize the speaker's pronunciation characteristics. Experimental results show that, comparing to state-of-the-art algorithms, the proposed AC-SOM algorithm can dramatically improve the training and recognition speed without significantly impacting the recognition rate. This study therefore provides a highly promising option for the implementation of edge intelligent speaker recognition systems on resource-constrained devices.

References

1. Kinnunen T, Li H (2010) An overview of text-independent speaker recognition: from features to supervectors. Speech Commun 52(1):12–40
2. Singh N, Khan RA, Shree R (2012) Applications of speaker recognition. Proced Eng 38(1):3122–3126
3. Daqrouq K, Tutunji TA (2015) Speaker identification using vowels features through a combined method of formants, wavelets, and neural network classifiers. Appl Soft Comput J 27(2):231–239
4. Ajmera PK, Jadhav DV, Holambe RS (2011) Text-independent speaker identification using radon and discrete cosine transforms based features from speech spectrogram. Pattern Recogn 44(10):2749–2759
5. Yu JC, Zhang RL (2009) Speaker recognition method using MFCC and LPCC features. Comput Eng Des 30(5):1189–1191
6. Hermansky H (1990) Perceptual linear predictive (PLP) analysis of speech. J Acoust Soc Am 87(4):1738
7. Tirumala SS, Shahamiri SR, Garhwal AS, Wang R (2017) Speaker identification features extraction methods: a systematic review. Expert Syst Appl 90(12):250–271
8. Visalakshi R, Dhanalakshmi P (2014) Acoustic feature extraction methods LPC, LPCC and RASTA-PLP in speaker recognition. Asian J Inf Technol 13(10):595–598
9. Joshi D, Nakamura BH, Hahn ME (2015) High energy spectrogram with integrated prior knowledge for EMG-based locomotion classification. Med Eng Phys 37(5):518–524
10. Li P, Zhang S, Feng H et al (2015) Speaker identification using spectrogram and learning vector quantization. J Comput Inf Syst 11(9):3087–3095
11. Liu Z, Wu Z, Li T et al (2018) GMM and CNN hybrid method for short utterance speaker recognition. IEEE Trans Industr Inf 43(99):11–17
12. Rajeswara Rao R, Prasad A, Kedari Rao Ch (2012) Robust features for automatic text-independent speaker recognition using ergodic Hidden Markov Models (HMMs). Digit Signal Process 4(3):24–33
13. Gupta M, Bharti SS, Agarwal S (2019) Gender-based speaker recognition from speech signals using GMM model [J]. Mod Phys Lett B 33(35):23–143
14. Kyung YJ, Lee HS (1999) Bootstrap and aggregating VQ classifier for speaker recognition. Electron Lett 35(12):973–974
15. Chang HY, Kong AL, Li H (2010) GMM-SVM kernel with a bhattacharyya-based distance for speaker recognition. IEEE Trans Audio Speech Lang Process 18(6):1300–1312
16. Rakhmanenko IA, Meshcheryakov RV (2017) Identification features analysis in speech data using GMM-UBM speaker verification system. Tr Spiiran 3(52):32–50
17. Ali H, Tran SN, Benetos E, Garcez ASD (2018) Speaker recognition with hybrid features from a deep belief network. Neural Comput Appl 29(6):13–19
18. Fred R, Douglas R, Najim D (2015) Deep neural network approaches to speaker and language recognition. IEEE Signal Process Lett 22(10):1671–1675
19. Bazrafkan S, Corcoran PM (2018) Pushing the AI envelope: merging deep networks to accelerate edge artificial intelligence in consumer electronics devices and systems. IEEE Consum Electr Mag 7(2):55–61
20. Zeng FZ, Zhou H (2013) Speaker recognition based on a novel hybrid algorithm. Proced Eng 61(1):220–226
21. Kovács G, Tóth L, Van CD et al (2017) Increasing the robustness of CNN acoustic models using autoregressive moving average spectrogram features and channel dropout. Pattern Recogn Lett 100(1):44–50
22. Sarlin P (2015) Automated and weighted self-organizing time maps. Knowl Inf Syst 44(2):493–505
23. Carboni OA, Russu P (2015) Assessing regional wellbeing in Italy: an application of malmquist–DEA and self-organizing map neural clustering. Soc Indic Res 122(3):677–700
24. Hinton G, Deng L, Yu D, Dahl GE et al (2012) Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process Mag 29(6):82–97
25. Cheng F, Wang SLLA (2018) Visual speaker authentication with random prompt texts by a dual-task CNN framework. Pattern Recogn 83(1):340–352
26. Ding S, Su C, Yu J (2011) An optimizing BP neural network algorithm based on genetic algorithm. Artif Intell Rev 36(2):153–162