TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 01/11 pp371–378
DOI: 10.26599/TST.2018.9010121
Volume 24, Number 4, August 2019

A Deep Learning Method for Chinese Singer Identification

Zebang Shen, Binbin Yong, Gaofeng Zhang, Rui Zhou, and Qingguo Zhou*

Abstract: As a subfield of Multimedia Information Retrieval (MIR), Singer IDentification (SID) is still in the research phase. On one hand, SID cannot easily achieve high accuracy because the singing voice is difficult to model and always disturbed by the background instrumental music. On the other hand, the performance of conventional machine learning methods is limited by the scale of the training dataset. This study proposes a new deep learning approach based on Long Short-Term Memory (LSTM) and Mel-Frequency Cepstral Coefficient (MFCC) features to identify the singer of a song in large datasets. The results of this study indicate that LSTM can be used to build a representation of the relationships between different MFCC frames. The experimental results show that the proposed method achieves better accuracy for Chinese SID in the MIR-1K dataset than the traditional approaches.

Key words: singer identification; timbre modeling; deep learning; long short-term memory

1 Introduction

The bulk of multimedia data on the Internet grows at 1000-fold speed. Hence, online multimedia content indexing and retrieval has become one of the most cutting-edge topics in the multimedia domain. To retrieve multimedia contents precisely and efficiently, extensive research has been conducted in the field of Multimedia Information Retrieval (MIR), which involves building a robust categorical and retrieval system[1]. Music is one of the most widely used multimedia contents on the Internet. Although retrieving a song using several keywords, such as singer name or song name, is easy, retrieving a song using a small piece of music can be fairly complex. To avoid retrieval dependency on keywords, various Content-Based Retrieval (CBR) techniques have been developed.

Singer IDentification (SID) is a subfield of CBR, which retrieves the singer name through a small piece of music. Therefore, SID has been used to classify and group massively disordered music data in the past years. Furthermore, accurate SID can be used in digital rights management when the singer of music on the Internet can be identified automatically. However, the SID application is still undeveloped at present. First, high accuracy is still difficult to achieve because the singing voice is different from the speaking voice. Hence, the problem is difficult to model. Meanwhile, the background instrumental music and audio of other singers may affect the accuracy of SID. Second, the performance of conventional machine learning methods is limited by the use of small-scale training dataset.

To solve these problems, various approaches have been proposed to identify the singer on the basis of the music content precisely. Many of the proposed approaches focus on the singer feature extraction. For example, Dupraz and Richard[2] investigated audio fingerprinting, which can be used in SID. Schindler and Rauber[3] combined several types of features to improve the identification accuracy. Cai et al.[4] analyzed the auditory sense features of humans and combined many single features to train a Gaussian mixture model. Patil et al.[5] used the Mel-Frequency Cepstral Coefficient (MFCC) features and cepstral mean subtracted features

• Zebang Shen, Binbin Yong, Gaofeng Zhang, Rui Zhou, and Qingguo Zhou are with the School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China. E-mail: shenzb12@lzu.edu.cn; yongbb14@lzu.edu.cn; zhanggaof@lzu.edu.cn; zr@lzu.edu.cn; zhouqg@lzu.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2018-07-12; accepted: 2018-09-02

to examine SID and achieved accuracy rates of 75.75% and 84.5% for a dataset of 500 songs, respectively. Several artificial intelligence methods have also been applied in the SID field, but most works use the fully connected Back Propagation Neural Network (BPNN)[6] and Support Vector Machine (SVM)[7,8]. The running speed of BPNN is fast, but the BPNN can be easily overfitted. SVM works well for small samples. However, for the large amount of music data on the Internet, the performance of SVM becomes worse. Moreover, single SVM aims for binary classification problems, but it is complex to construct the model with SVM for multi-class classification problems. When the number of classes is large, the training time of SVM is unacceptable. To solve the second problem of the SID application, we need more efficient approaches.

To apply SID to a large-scale dataset, a deep learning based method is proposed in this study. Since the pioneering work of Hinton and Salakhutdinov[9], deep learning has revolutionized many fields. One successful application of deep learning is speech recognition[10]. By using Deep Neural Networks (DNNs)[11] or deep Recurrent Neural Networks (RNNs)[12], we can build complex acoustic models that map the input features to grapheme outputs correctly. However, in the field of MIR, particularly the SID area, research based on deep learning is still rare. Shen et al.[13] used Long Short-Term Memory (LSTM) to do SID in their study. Inspired by the successful application of RNNs on acoustic modeling, a new deep learning method for SID, which is based on the multilayer LSTM networks, is presented in this study. In addition, several regularization strategies have been introduced to control overfitting.

The remainder of this paper is organized as follows: Section 2 gives a brief introduction to the RNNs and LSTM networks. Section 3 describes the architecture of the proposed method. Section 4 discusses the settings of the experiments. Section 5 provides the experimental results and analysis. Finally, Section 6 draws the conclusion.

2 Introduction to LSTM

The LSTM[14] is a type of RNN with loops and gates. The LSTM network is the basic layer of the proposed model. By stacking several LSTM layers and combining the LSTM networks with DNNs, we construct the SID deep learning model.

2.1 Recurrent neural networks

The RNN is a neural network that is designed specifically for processing a sequence of values $(x^{(1)}, ..., x^{(\tau)})$, which is based on an early idea proposed in machine learning and statistical models: sharing parameters across different parts of a model[15]. Similar to convolutional neural networks that share weights across pixels, RNNs share weights across time steps. Figure 1 shows the computational graph and unfolding state of a commom RNN structure.

For a static length of series data $x^{(1)}, x^{(2)}, \ldots, x^{(\tau)}$, RNNs perform forward propagation to calculate the loss and backpropagation to update the weights by gradient descent across the time steps. First, the hidden layer inputs at time t, which are related to the current inputs and the hidden states at time $t-1$, are calculated using Eq. (1).

$$a^{(t)} = b + Wh^{(t-1)} + Mx^{(t)} \tag{1}$$

where M is the input weight matrices and W is the hidden layer weight matrices, $h^{(t-1)}$ denotes the hidden state at time step $t-1$, $x^{(t)}$ denotes the input data at time t, and b is the hidden layer biases. Then, the hidden states at time t are calculated using the nonlinear activation function.

$$h^{(t)} = \tanh(a^{(t)}) \tag{2}$$

The output of the hidden layer $o^{(t)}$ is formally written as

$$o^{(t)} = c + Nh^{(t)} \tag{3}$$

where N is the output weight matrices and c is the output biases. Afterward, the softmax is performed on the outputs.

$$p^{(t)} = \text{softmax}(o^{(t)}) \tag{4}$$

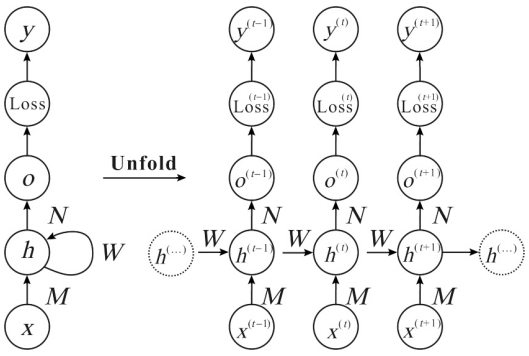


Fig. 1 Computational graph and its unfolding state of RNNs. x denotes the input sequence, h is the hidden state, Loss measures how far the output o is from the training target y, and M, N, and W are the weights of RNNs.

in which $p^{(t)}$ is the prediction of the input at time t with respect to the previous data because the RNN is an iterative model. Then, the cross entropy between the predictions and actual labels is calculated as the loss function.

$$\text{Loss} = -\sum_t y^{(t)} \log(p^{(t)}) \tag{5}$$

where $y^{(t)}$ is the actual label at time t. Finally, the gradient descent algorithm based on Loss is utilized to train the RNN.

RNNs share parameters along the time axis, which enable the past states of the model to influence the current states. For SID, this type of neural network is very appropriate because the audio is a continuous sequence. However, the gradient vanishing problem will occur when the sequence is long. That is, the gradients reach 0 if the sequence is long, which leads to training failure. To overcome this problem, the LSTM has been proposed.

2.2 Long short-term memory

Given that the LSTM network is a kind of RNN, its basic structure is similar to that of RNNs. To overcome the graident vanishing problem, LSTM uses the LSTM cell instead of hidden neurons in RNNs. The structure of the LSTM cell is shown in Fig. 2. The LSTM cell contains the forget, input, and output gates[16], which can be expressed as three functions, namely, $f^{(t)}$, $i^{(t)}$, and $o^{(t)}$. $f^{(t)}$ decides which information model should be discarded from the cell state. The equation of the forget gate is expressed as

$$f^{(t)} = \sigma_g(W_f x^{(t)} + V_f h^{(t-1)} + b_f) \tag{6}$$

To simplify the sign, W and V in the equations denote the weight matrices and b denotes the biases of the model. These weights and biases are the parameters that should be learned during training. $x^{(t)}$ is the input vector at time t and $h^{(t)}$ refers to the output vector of the cell at time t. σ_g is the sigmoid function which can be defined as

$$\sigma_g(x) = \frac{e^x}{e^x + 1} \tag{7}$$

The input gate decides which information should be stored in the cell states, whereas the output gate decides which information should be the output. The equations of the input gate $i^{(t)}$ and output gate $o^{(t)}$ are defined as follows:

$$i^{(t)} = \sigma_g(W_i x^{(t)} + V_i h^{(t-1)} + b_i) \tag{8}$$

$$o^{(t)} = \sigma_g(W_o x^{(t)} + V_o h^{(t-1)} + b_o) \tag{9}$$

The cell state $c^{(t)}$ is updated as follows:

$$c^{(t)} = f^{(t)} \circ c^{(t-1)} + i^{(t)} \circ \sigma_h(W_c x^{(t)} + V_c h^{(t-1)} + b_c) \tag{10}$$

The operator $\circ$ denotes the Hadamard product and σ_h refers to the hyperbolic tangent function.

$$\sigma_h(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{11}$$

The output of the cell at time t is

$$h^{(t)} = o^{(t)} \circ \sigma_h(c^{(t)}) \tag{12}$$

Through the LSTM cell, the LSTM networks can deal with the long-term dependency of a long sequence without gradient exploding or vanishing. Except for the internal computation, the forward and backpropagation of LSTM are much similar to those of RNNs. An advantage of LSTM over conventional RNNs, hidden Markov models, and other sequence machine learning methods is its relative insensitivity. Thus, in this study, the LSTM neural network is adopted in conducting SID because the singing voice is a typical sequence data with long dependency.

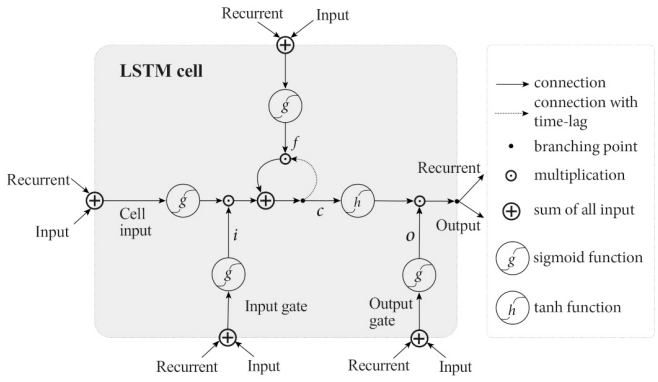


Fig. 2 LSTM cell.

3 Proposed Method for Singer Identification

In this study, we use the MIR-1K dataset, which contains 1000 segments of Chinese popular music with a fixed sampling rate of 22 050 Hz, to conduct the experiments. The audios in the MIR-1K dataset contain several nonvocal parts. Therefore, we cut these nonvocal parts manually. Then, the MFCC features are extracted and rescaled to zero mean and unit variance. Although MFCC feature extraction used sliding windows on the original audios, in our method, we use a sliding window with a fixed size to select a fixed-length sequence from the MFCC features. Actually, this window size is the number of time steps of the input sequences. Then, the proposed model is trained with these sequences using the Adam optimization algorithm[17]. After training, the model is

tested with a test set, which is another subset of the MIR-1K dataset. The complete model is illustrated in Fig. 3.

3.1 MFCC features

MFCC features[18,19] are widely used in automatic speech and speaker recognition. The MFCC is a representation of the short-term power spectrum of a sound which is based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency[20]. Essentially, the MFCCs are coefficients that collectively comprise a Mel-Frequency Cepstrum (MFC). In the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly spaced frequency bands used in the normal cepstrum. Thus, the MFCCs are considered to be appropriate for SID.

To get the MFCCs features, the original signal is framed into 40 ms short frames. Given that the sampling rate of the dataset is 22 050 Hz, the frame length is

$$0.04 \times 22\,050 = 882 \tag{13}$$

The frame shift is set at 512 in our method. Then 20 and 40 coefficients are extracted from each frame for comparison. After feature extraction, these MFCC spectra are standardized and rescaled to zero mean and unit variance.

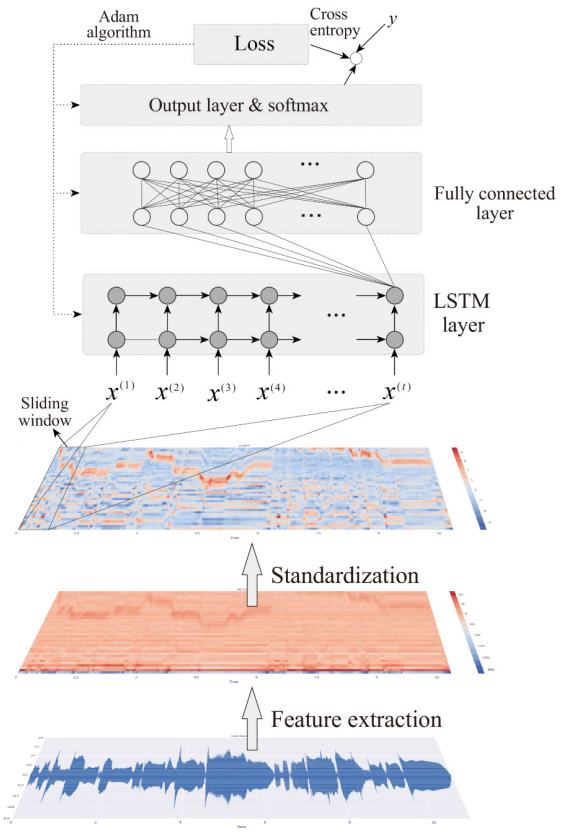


Fig. 3 Complete processing flow of the proposed method.

After standardization, the feature sequences of the audio are still too long for training a model. Thus, these long sequences should be split into a set of small sequences. Instead of using equal-length sequences, the sliding window with a fixed window size is adopted in our method. For comparison purposes, we select the time steps of 10 and 20 at different stages of the experiments. For the 20-step sequence, the audio duration is only 0.5 s, which is smaller than that of previous works[1,4,5,21,22]. Furthermore, these sequences are shuffled and inputted into the model for training.

3.2 Model architecture

As shown in Fig. 3, the proposed model is a DNN, which consists of two LSTM layers, one fully connected layer, and one ouput layer. The numbers of hidden neurons in the two LSTM layers are both 400, which is confirmed by the experimental results presented in Section 5. The outputs of the last time step in the second LSTM layer are the inputs of the fully connected layer.

After the LSTM layers, the model has a fully connected layer with 400 hidden neurons. The activation function in this layer is the rectified linear units[23]. Finally, the model has an output layer, which maps the outputs of the fully connected layer to 10 units and calculates the softmax. The cross entropy is selected as the loss function, and the formula has been expressed as Eq. (5).

To start training, the parameters of the proposed model are initialized with a zero mean and unit variance Gaussian distribution. The Adam optimization algorithm is used to train the DNN. The Adam optimization algorithm is based on adaptive estimates of lower-order moments and well suited for problems with a large amount of data and many parameters. The deep learning model for real-world SID application is a large model with many layers and parameters. Thus, we use the Adam optimization algorithm for training.

To overcome the overfitting problem, the dropout[24] and early stopping[25] strategies are adopted. For each small batch, the units in the LSTM layers and fully connected layers are randomly dropped out with a fixed probability of 50%. This regularization method is

widely used, thus, we will not discuss it in detail.

The early stopping strategy, which is also used to prevent overfitting, is shown in Algorithm 1. A validation set is used to validate the cost of every 100 batches while training. The early stopping method helps stop the training process before overfitting occurs.

4 Experimental Setup

To verify the performance of the proposed method, a series of experiments is conducted. This section describes the experimental setup.

4.1 Dataset description

We use a subset of the MIR-1K dataset because the data in each category of the original MIR-1K dataset are uneven. Several singers have 132 song segments, whereas other singers have only 8 song segments. Thus, a subset of 10 singers with 400 song segments is selected. In this subset, each singer sings 5 complete songs that are split into 40 segments. The details of the subset are given in Table 1. The validation set is a small randomly selected subset of that is not used for training.

Algorithm 1 Early stopping algorithm in proposed method

Input: training dataset X and validation set V
Output: model M
Set: $patiance \leftarrow 10$ & current patiance $c = 0$
  initialize M
  while $c < patiance$ **do**
    train M with Adam algorithm
    if $step\ \%100 == 0$ **then**
      if cost of V is smaller **then**
        save the copy of M as M'
      else
        $c++$
  return the latest copy M'

Table 1 Details of the dataset.

Singer name	Gender	Total song (segment)	Training song (segment)	Testing song (segment)
Abjones	Male	5 (40)	4 (31)	1 (9)
Annar	Female	5 (40)	4 (31)	1 (9)
Bobon	Male	5 (40)	4 (32)	1 (8)
Fdps	Male	5 (40)	4 (31)	1 (9)
Heycat	Female	5 (40)	4 (32)	1 (8)
Jmzen	Male	5 (40)	4 (32)	1 (8)
Kenshin	Male	5 (40)	4 (32)	1 (8)
Stool	Male	5 (40)	4 (32)	1 (8)
Titon	Female	5 (40)	4 (32)	1 (8)
Yifen	Female	5 (40)	4 (32)	1 (8)

4.2 Feature extraction

After MFCC feature extraction, a sliding window is used to split these feature spectra into a set of sequence data. We use different window sizes here for different experiments. The details of final input data are shown in Tables 2 and 3.

4.3 Performance metrics

To measure the performance of the experimental models, accuracy (AC), precision (PR), recall (RC), and F-Measure (FM) are calculated for each model. For comparison, contrast experiments have been conducted with three machine learning methods, namely, SVM, Deep Fully Connected Neural Network (DFCNN), and Random Forest (RF). These methods are widely used in previous works on SID[1, 7, 8] and are appropriate as the performance benchmarks.

5 Results

Table 4 shows the experimental results of the MIR-1K

Table 2 Details of the input data with 20 MFCC features.

Window size	Time step	Training set size	Test set size	Size of one input
10	10	(78753, 10, 20)	(20675, 10, 20)	(10, 20)
20	20	(75583, 20, 20)	(19845, 20, 20)	(20, 20)

Table 3 Details of the input data with 40 MFCC features.

Window size	Time step	Training set size	Test set size	Size of one input
10	10	(78753, 10, 40)	(20675, 10, 40)	(10, 40)
20	20	(75583, 20, 40)	(19845, 20, 40)	(20, 40)

Table 4 The performance of the proposed method and the benchmarks.

Feature and step	Method	AC (%)	PR (%)	RC (%)	FM (%)
20 features 10 steps	RF	67.8	70	68	68
	SVM	81.5	82	82	82
	DFCNN	78.6	79	79	79
	LSTM	**86.2**	**86**	**86**	**86**
20 features 20 steps	RF	68.6	71	69	68
	SVM	82.2	82	82	82
	DFCNN	81.1	82	81	81
	LSTM	**88.3**	**89**	**88**	**88**
40 features 10 steps	RF	70.6	73	71	71
	SVM	78.6	79	79	79
	DFCNN	81.0	81	81	81
	LSTM	**85.4**	**86**	**85**	**85**
40 features 20 steps	RF	70.6	73	71	71
	SVM	79.1	79	79	79
	DFCNN	81.7	82	82	82
	LSTM	**88.4**	**89**	**88**	**88**

dataset, from which we can observe that the proposed method outperforms the conventional machine learning methods, particularly when more features and time steps are utilized. The best accuracy of the proposed method for SID is approximately 88.4% for the audio duration of only 0.5 s. The optimized SVM achieves the second-best accuracy when the data and time steps are small. However, the use of SVM for real-world SID application is expensive. SID in the music information retrieval system has to deal with a large amount of data with numerous classes, in which case the training time of the SVM classifier will be unacceptable.

The DFCNN with two hidden layers and Adam optimization algorithm also achieves the second-best accuracy when the numbers of features and time steps are large. The DFCNN can build complex representations by increasing the number of hidden layers and neurons. The proposed method is essentially a DFCNN. Instead of totally using a fully connected layer, we use the LSTM layer, which considers parameter sharing and long-term dependency. This change helps improve the identification accuracy considerably without the loss of efficiency.

By comparing the proposed method with different lengths of time steps (the sequence length), we observe two interesting phenomena: first, the human timbre has long-term dependency; second, the LSTM neural network can model this type of dependency. For the first phenomenon, according to previous works[1] and our experiments, the longer the sequence is, the higher the accuracy that the model can achieve. A long and continuous audio contains more information than a few discrete frames of features, which is the same for humans as we can always recognize a singer with a long piece of music. The information that exists between continuous frames can be considered long-term dependency. For the second phenomenon, the identification accuracy of the LSTM-based method increases with the number of time steps, which indicates that the LSTM can model this type of long-term dependency. By contrast, the number of MFCC features is unimportant for SID, and the results of 20 and 40 features are similar.

To analyze the influence of the number of hidden neurons on the identification accuracy of the proposed method, we compare the proposed method with different hidden neurons in the same dataset. The results are shown in Table 5. In this experiment, 40 MFCC features and 20 time steps are selected. The proposed method with 400 hidden neurons achieves the highest accuracy. When the number of hidden neurons is more than 400, the performance is not improved, whereas the computation time increases. Thus, the hidden LSTM layer with 400 neurons is selected as the final model.

Table 5 Comparison of the proposed methods with different numbers of hidden neurons.

Model	AC (%)	PR (%)	RC (%)	FM (%)
LSTM with 100 hidden neurons	85.3	86	85	85
LSTM with 200 hidden neurons	87.2	87	86	88
LSTM with 300 hidden neurons	88.2	88	88	88
LSTM with 400 hidden neurons	88.4	89	88	88
LSTM with 500 hidden neurons	88.3	89	88	88

6 Conclusion

The results show that the proposed deep learning method improves the performance of Chinese SID on the MIR-1K dataset compared with the conventional machine learning methods, such as SVM, RF, and deep feedforward network with fully connected structure. Hence, deep learning methods should be introduced to the field of SID when developing a real-world SID application. In contrast to that proposed in previous works, the proposed deep learning method can deal with a large dataset that contains thousands of classes and numerous sample audios by increasing the model's hidden neurons and LSTM layers.

Although the proposed method outperformed the traditional approaches, developing a real-world SID application by employing the dataset used in this study is still inadequate. In the future, we can build a public SID dataset with numerous sample audios and thousands of classes. On the basis of the public dataset, more deep learning based SID approaches could be investigated, which helps develop automatic SID applications for large-scale dataset. To speed up model inference, several edge computing techniques, such as fog computing, mobile edge computing, and dew computing, can be applied in the future[26].

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 61402210 and 60973137), the Program for New Century Excellent Talents in University (No. NCET-12-0250), the Major Project of High

Resolution Earth Observation System (No. 30-Y20A34-9010-15/17), the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDA03030100), the Gansu Sci. & Tech. Program (Nos. 1104GKCA049, 1204GKCA061, and 1304GKCA018), the Fundamental Research Funds for the Central Universities (No. lzujbky-2016-140), and Google Research Awards and Google Faculty Research Awards, China. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Jetson TX1 used for this research.

References

[1] S. Masood, J. S. Nayal, and R. K. Jain, Singer identification in Indian Hindi songs using MFCC and spectral features, in *Proc. IEEE 1st Int. Conf. Power Electronics, Intelligent Control and Energy Systems*, Delhi, India, 2016, pp. 1–5.

[2] E. Dupraz and G. Richard, Robust frequency-based audio fingerprinting, in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Dallas, TX, USA, 2010, pp. 281–284.

[3] A. Schindler and A. Rauber, A music video information retrieval approach to artist identification, in *Proc. 10th Int. Symp. Computer Music Multidisciplinary Research*, Marseille, France, 2013.

[4] W. Cai, Q. Li, and X. Guan, Automatic singer identification based on auditory features, in *Proc. 7th Int. Conf. Natural Computation*, Shanghai, China, 2011, pp. 1624–1628.

[5] H. A. Patil, P. G. Radadia, and T. K. Basu, Combining evidences from mel cepstral features and cepstral mean subtracted features for singer identification, in *Proc. Int. Conf. Asian Language Processing*, Hanoi, Vietnam, 2012, pp. 145–148.

[6] B. Whitman, G. Flake, and S. Lawrence, Artist detection in music with Minnowmatch, in *Proc. IEEE Workshop on Neural Networks for Signal Processing*, North Falmouth, MA, USA, 2001, pp. 559–568.

[7] N. C. Maddage, C. S. Xu, and Y. Wang, Singer identification based on vocal and instrumental models, in *Proc. 17th Int. Conf. Pattern Recognition*, Cambridge, UK, 2004, pp. 375–378.

[8] Y. E. Kim and B. Whitman, Singer identification in popular music recordings using voice coding features, in *Proc. 3rd Int. Conf. Music Information Retrieval*, Paris, France, 2002, pp. 164–169.

[9] G. E. Hinton and R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[10] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] G. Hinton, L. Deng, D. Yu, G. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[12] A. Graves, A. R. Mohamed, and G. Hinton, Speech recognition with deep recurrent neural networks, in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Vancouver, Canada, 2013, pp. 6645–6649.

[13] Z. Shen, B. Yong, G. Zhang, R. Zhou, and Q. Zhou, A deep learning method for Chinese singer identification, in *Sixth International Conference on Advanced Cloud and Big Data*, Lanzhou, China, 2018.

[14] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[16] F. A. Gers, J. Schmidhuber, and F. Cummins, Learning to forget: Continual prediction with LSTM, *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[17] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.

[18] P. Mermelstein, Distance measures for speech recognition, psychological and instrumental, in *Pattern Recognition and Artificial Intelligence*, R. C. H. Chen, ed. Academic Press, 1976, pp. 374–388.

[19] S. Davis and P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[20] M. Sahidullah and G. Saha, Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition, *Speech Communication*, vol. 54, no. 4, pp. 543–565, 2012.

[21] T. Zhang, Automatic singer identification, in *Proc. 2003 Int. Conf. Multimedia and Expo*, Baltimore, MD, USA, 2003, pp. 1–33.

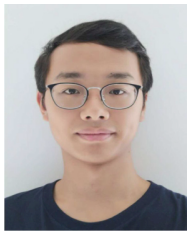
[22] Y. Hu and G. Z. Liu, Automatic singer identification using missing feature methods, in *Proc. IEEE Int. Conf. Multimedia and Expo*, San Jose, CA, USA, 2013, pp. 1–6.

[23] X. Glorot, A. Bordes, and Y. Bengio, Deep sparse rectifier neural networks, in *Proc. 14th Int. Conf. Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, 2011, pp. 315–323.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] L. Prechelt, Automatic early stopping using cross validation: Quantifying the criteria, *Neural Networks*, vol. 11, no. 4, pp. 761–767, 1998.

[26] Y. Z. Zhou, D. Zhang, and N. X. Xiong, Post-cloud computing paradigms: A survey and comparison, *Tsinghua Science and Technology*, vol. 22, no. 6, pp. 714–732, 2017.



Zebang Shen received the bachelor degree from Lanzhou University in 2016. Now he is a master student in Lanzhou University. He is researching in artificial neural network, machine learning, deep learning, and big data.

Binbin Yong received the master degree from Lanzhou University in 2012, and received the PhD degree from Lanzhou University in 2017. He is researching in parallel computing of GPU, machine learning, deep learning, and general vector machine.

Gaofeng Zhang received the MS degree from Southwest Jiaotong University. Now he is a lecturer in the School of Information Science and Technology, Lanzhou University. His research interests are deep learning, renormalization group, and computer simulation.

Rui Zhou received the BS and PhD degrees from Lanzhou University in 2004 and 2010, respectively. Now he is an associate professor who is researching embedded systems and real-time systems in School of Information Science & Engineering, Lanzhou University.

Qingguo Zhou received the BS, MS, and PhD degrees from Lanzhou University in 1996, 2001, and 2005, respectively. Now he is a professor of the School of Information Science and Engineering, Lanzhou University. He is also a fellow of IET. He was a recipient of IBM Real-Time Innovation Award in 2007, Google Faculty Award in 2011, and Google Faculty Research Award in 2012. His research interests include safety-critical systems, embedded systems, and real-time systems.