A ROBUST TEXT-INDEPENDENT SPEAKER VERIFICATION METHOD BASED ON SPEECH SEPARATION AND DEEP SPEAKER

Fei Zhao, Hao Li, Xueliang Zhang

College of Computer Science, Inner Mongolia University, China

{zhaofei_works,lihao.0214}@163.com,cszxl@imu.edu.cn

ABSTRACT

Recently, deep neural networks (DNNs) have achieved incredible performance in speaker verification. However, most of which remains sensitive to environment noise. In this paper, we propose an end-to-end speaker verification framework to enhance the robustness against background noise. The proposed framework first utilizes convolutional recurrent network (CRN) to address speech separation. Then the output of the middle layer of the CRN is used as the auxiliary feature, and together with the robust Filter banks (Fbanks) feature of noisy speech are fed to the speaker verification system. The speech separation and speaker verification are jointly optimized. Compared with deep speaker and DNN/i-vector, systematic evaluation indicates that the proposed algorithm can obtain a better performance in noisy conditions.

Index Terms— Robust speaker verification, Speech separation, Convolutional recurrent network, Deep speaker

1. INTRODUCTION

Speaker verification (SV) is a task of judging whether it is a declared speaker identity through the information of the speech. Depending on whether the speech content of enrolling and testing are the same, it can be divided into text-dependent SV [1] and text-independent SV [2, 3]. Apparently, with no limitation on content during test, the text independent SV is more friendly to users. At the same time, it's also more difficult than text-dependent SV. This work focuses on text-independent SV.

I-vector [4] is a well known method which greatly improved the performance of SV. The method consists of several steps:

- Firstly, training a universal background model (UBM) [5] with a large amount of speech to collect sufficient statistics for extracting i-vector.
- Secondly, extract speaker i-vector, so hight-dimension statistics can converted into a single low-dimensional i-vector that representing the identity of the speaker.
- Finally, training probabilistic linear discriminant analysis (PLDA) [6] model, and produce verification scores by calculating distance between i-vector from different utterances.

Influenced by DNN powerful modeling ability and its successful application in automatic speech recognition (ASR) [7], Lei *et al* [8] used DNN to replace the gaussian mixture model (GMM) for acoustic modeling to extract i-vector. DNN can directly model the phoneme state space instead of the complex acoustic space, and significant improvements over the traditional GMM. Another effective technique is to use DNN to extract deep bottleneck features [9, 10] or obtain speaker representations directly [11, 12]. Driven by big data and increased computing power, end-to-end SV [12, 13, 14] can achieve better performance than classic i-vector approach. The output of the neural network is low-dimensional vector called embedding (also known as d-vector) which is adopted to represent the speaker identity.

Although research on SV has achieved big progress, noise is still a inevitable factor in real environment that impairs the performance of SV systems. A common strategy is using a frontend processing method to enhance both training and test set first, and conducting SV system on the enhanced training set. It may be able to improve the SV performance since the features may become cleaner after enhancement. However, the performance of this approach is highly dependent on the performance of the separation frontend [15].

More recently, Tan and Wang [16] incorporated a convolutional encoder-decoder (CED) and long short-term memory (LSTM) into the CRN architecture for speech separation. We speculate that the low-dimensional output of LSTM in CRN can be used as robust bottleneck features in SV system. In this paper, we integrate the speech separation into end-to-end speaker verification system, and jointly optimize the two modules both of which are based deep learning. Experimental results show that the proposed method outperforms the recent proposed end-to-end SV method deep speaker [12] and

The first two authors contributed equally to this work.

DNN/i-vector [8].

The rest of this paper is organized as follows. Section 2 briefly reviews the speech separation, end-to-end SV framework and triplet loss. Section 3 describes the framework proposed. Experiments and analysis are presented in the Section 4. Finally, summarize in Section 5.

2. RELATED WORK

2.1. Speech Separation

The purpose of speech separation is to extract the target speech from background interference [16, 17, 18, 19]. In recent years, mapping-based supervised speech separation have been proposed successively [20, 21], and achieves very promising separation performance in both matched and unmatched test conditions. For supervised speech separation, the usually input feature is magnitude spectrum. Besides, power spectrum, or other forms of spectra such as mel spectrum, was also used instead of magnitude spectrum. A log operation is usually applied to compress the dynamic range and facilitate training. In terms of cost function, mean squared error (MSE) is usually used. The loss function is given by:

$$L_{ss} = \frac{1}{N}\sum_{i=1}^{N}\left\|Y_i - \hat{Y}_i\right\|^2 \quad (1)$$

where N is the number of T-F unit, Y_i and $\hat{Y}_i$ represent the short-time Fourier transform (STFT) of pure speech and estimated under the i-th T-F unit, respectively.

The speech separation structure used in this study is based on CRN which is proposed in [16]. The CRN leads to consistently better objective speech intelligibility and quality compared with LSTM model [22]. Moreover, the CRN has much fewer trainable parameters. The structure of CRN is shown in Figure 1 (speech separation part). We use the magnitude spectrum of the mixture as the input feature. To compress the dynamic range of the value, a cubic root compression is applied. Then, the feature is normalized by zero mean and unit variance.

2.2. End-to-end Speaker Verification

End-to-end SV system has many types of architecture [1, 2, 12, 13]. Heigold *et al.* [1] used the last frame output of the LSTM as the utterance-level embedding for SV. In [14], Snyder *et al.* utilized a type of network-in-network (NIN) nonlinearity to form the speaker embedding. In this study, deep residual convolutional neural network (ResCNN) is used for SV as shown in Figure 1. For deep ResCNN, the CNN is effective for reducing spectral variations and modeling spectral correlations in acoustic features [23], and the deep network can better represent long utterance than shallow networks. The triplet loss based end-to-end SV architecture is shown in Figure 2. In the training stage, the frame embedding output from the deep model through pooling layer to form the utterance embedding and then normalized to the unit hyper-sphere through L2 normalize. In the evaluation stage, enrolled utterance embeddings from the same speaker are averaged to get speaker embeddings. Euclidean distance between enroll speaker embeddings and test utterance embeddings are calculated, which can be utilized for the final speaker verification decision.

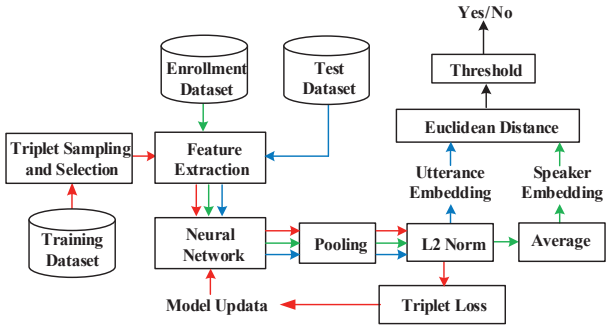

Fig. 2: End-to-end speaker verification system architecture based triplet loss.

Triplet loss [24] takes three samples as input, an anchor (an utterance from a specific speaker), a positive example (another utterance from the same speaker), and a negative example (an utterance from another speaker). Aiming to minimize the within-class distance and maximize the between-class distance. The loss L_{ds} for M triplets is defined as:

$$L_{ds} = \sum_{i=1}^{M}[E_i^{ap} - E_i^{an} + \alpha]_+ \quad (2)$$

where E_i^{ap} and E_i^{an} represents the Euclidean distance between anchor positive and anchor negative, respectively, α is an empirical value used to force a limit between the two distances. The operator $[x]_+$ =max(x,0) represents triplet selection.

3. JOINT TRAINING

3.1. Joint training

As illustrated in Figure 1, the key idea for joint training is to concatenate a deep ResCNN-based speaker verification and a CRN-based speech separation to form a larger and deeper neural network. The output of the LSTM in CRN is used as auxiliary feature, and together with the robust Fbank feature of noisy fed to the speaker verification model to estimate the speaker identity information. In training phase, the weights in all modules are jointly adjusted. The loss function of the

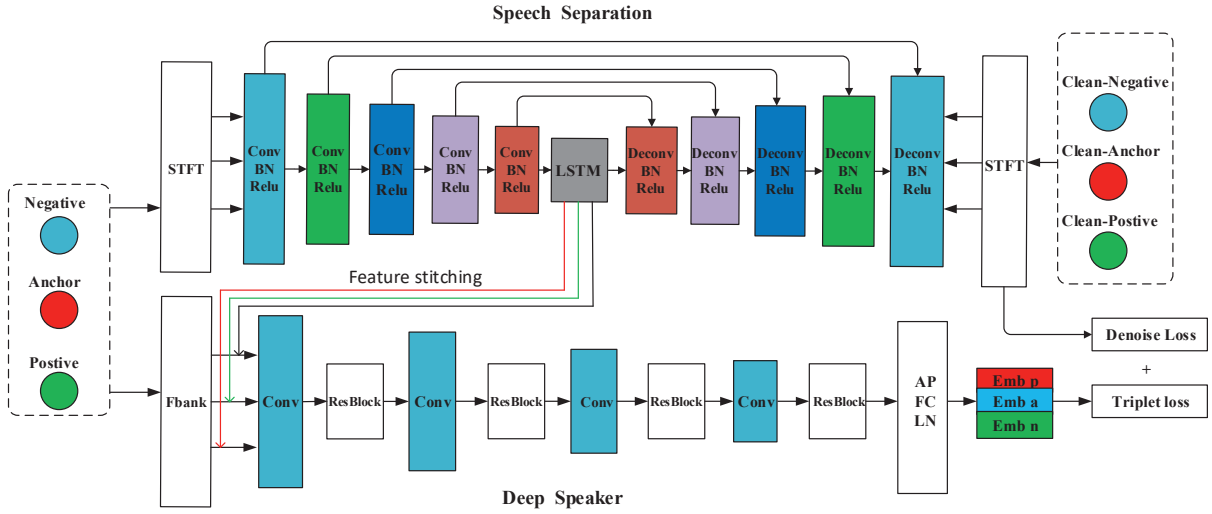

Fig. 1: Network architecture of our proposed speaker verification architecture.

speech separation and the deep speaker are employed together as the loss function of joint training architecture. The loss function can be written as:

$$L_{jt} = \beta L_{ss} + L_{ds} \tag{3}$$

where β is a weighting factor to adjust the trade-off between losses. L_{ss} represent the loss function of the speech separation part (E.q 1). L_{ds} represent the loss function of the deep speaker part (E.q 2).

3.2. Network architecture

The proposed network input is encoded into a low dimension latent space by several convolutional layers and then the following LSTM models the sequential information of the latent feature. The output of the LSTM is converted back to the original input shape by the decoder. CRN is recently invented architecture which combines the feature extraction capability of CNNs and the temporal modeling capability of recurrent neural networks (RNNs). The output of the LSTM is used as auxiliary feature for SV. For SV part, we use ResCNN architecture which contains 4 convolution layers, 4 residual blocks (ResBlocks), 1 average pooling (AP) layer, 1 fully connected layer (FC) and length normalization (LN) to produce utterance-level embedding.

A more detailed description of the architecture is provided in Table 1. The input size and the output size of each layer are specified in *featureMaps* $\times$ *timeSteps* $\times$ *frequencyChannels* format. The layer hyper-parameters are given as (*kernelSize*, *strides*, *outChannels*) for convolution and deconvolution layers. In speech separation part, the kernel size is 1$\times$3 (*Time*$\times$*Frequency*), the stride length is 1$\times$2 (*Time*,*Frequency*). We do not apply padding on time or frequency. The number of feature maps in each decoder layer is doubled by the skip connections. In SV part, a basic ResBlock layer is added to each adjacent convolutional and deconvolutional layers.

4. EXPERRIMENTS

4.1. Experimental setup

We use 797 female speakers to evaluate the experiment. Among the speakers, 402 are from NIST SRE 2006 (8conv condition) [25] and 395 are from NIST SRE 2008 (8conv condition) [26]. For each target speaker, eight two-channel telephone conversations are provided, and each conversation is about two minutes. For each utterance, the large chunks of silence are removed by voice activity detect technology. Utterances are then mixed with babble or speech-shaped noise (SSN) at signal-to-noise ratios (SNRs) of {-5, 0, 5, 10} to produce the noisy utterances. Each noise is about four minutes and is divided into two non-overlapping time portions. The first and the second parts are used for training and testing, respectively. In addition, to test the generalization performance of the proposed model, the SNRs of {-3, 3, 8, 12} are also involved as SNR-unmatch condition.

The proposed method and deep speaker are implemented by using open-source AI framework PyTorch [27]. For the proposed method, the parameter β is tuned to balance the two training losses and set to 0.1 according to our experiments. The models are trained with Adam optimizer [28]. We set learning ration to 0.001. Margin α is set to 0.1 and a mini-batch size of 64.

Equal error rate (EER) is utilized as evaluation indicator

Table 1: Architecture of the proposed method. Here T denotes the number of frames in the *STFT* magnitude spectrum.

Component	Layer	Hyperparameters	Output size
Speech Separation	reshape_1	-	$1\times T\times 161$
	conv2d_1	1×3,(1,2),8	$8\times T\times 80$
	conv2d_2	1×3,(1,2),8	$8\times T\times 39$
	conv2d_3	1×3,(1,2),16	$8\times T\times 19$
	conv2d_4	1×3,(1,2),16	$16\times T\times 9$
	conv2d_5	1×3,(1,2),16	$16\times T\times 4$
	reshape_2	-	$T\times 64$
	lstm_1	64	$T\times 64$
	lstm_2	64	$T\times 64$
	deconv2d_1	1×3,(1,2),16	$16\times T\times 9$
	deconv2d_2	1×3,(1,2),16	$16\times T\times 19$
	deconv2d_3	1×3,(1,2),8	$8\times T\times 39$
	deconv2d_4	1×3,(1,2),8	$8\times T\times 80$
	deconv2d_5	1×3,(1,2),1	$1\times T\times 161$
Deep Speaker	reshape_1	-	$1\times 200\times 128$
	conv2d_1	5×5,(2,2),(2,2),16	$16\times 100\times 64$
	Res_1	[3×3,16] [3×3,16]	$16\times 100\times 64$
	conv2d_2	5×5,(2,2),(2,2),64	$64\times 50\times 32$
	Res_2	[3×3,64] [3×3,64]	$64\times 50\times 32$
	conv2d_3	5×5,(2,2),(2,2),128	$128\times 25\times 16$
	Res_3	[3×3,128] [3×3,128]	$128\times 25\times 16$
	conv2d_4	5×5,(2,2),(2,2),256	$256\times 13\times 8$
	Res_4	[3×3,256] [3×3,256]	$256\times 13\times 8$
	AP	-	$256\times 4\times 1$
	reshape_2	-	1024
	FC	-	512
	LN	-	512

to measure the overall performance of the system. The EER represents the value at which the false positive rate equals to the false negative rate in test set.

Table 2: EER (%) under matched SNR conditions with SSN

SNR	-5dB	0dB	5dB	10dB	Ave
System	EER	EER	EER	EER	EER
1) GMM/i-vector	11.26	7.37	6.06	5.93	7.65
2) DNN/i-vector	10.87	**5.97**	5.59	5.74	7.04
3) Deep Speaker	8.19	6.64	5.61	5.52	6.37
4) Proposed	**7.53**	6.02	**5.41**	**5.21**	**6.04**

4.2. Experimental results

We compare the proposed method with GMM/i-vector, DNN/i-vector and deep speaker. Table 2 shows the results with SSN under the matched SNR condition. Compared with DNN/i-vector and deep speaker the proposed method obtains 14.21% (from 7.04% to 6.04%) and 5.18% (from 6.37% to 6.04%) relative improvement for EER, respectively. For the babble noise, we can see similar results, as shown in Table 3, that the improvements are about 22.21% and 0.86% compared with DNN/i-vector and deep speaker, respectively.

Table 3: EER (%) under matched SNR conditions with babble

SNR	-5dB	0dB	5dB	10dB	Ave
System	EER	EER	EER	EER	EER
1) GMM/i-vector	13.23	9.01	7	5.8	8.76
2) DNN/i-vector	11.76	6.44	5.84	5.67	7.43
3) Deep Speaker	7.05	5.84	**5.27**	**5.14**	5.83
4) Proposed	**6.60**	**5.67**	5.5	5.35	**5.78**

Table 4: EER (%) under unmatched SNR conditions with SSN

SNR	-3dB	3dB	8dB	12dB	Ave
System	EER	EER	EER	EER	EER
1) GMM/i-vector	11.47	6.75	5.97	5.67	7.47
2) DNN/i-vector	11.38	5.9	5.78	**5.03**	7.02
3) Deep Speaker	7.35	5.88	5.37	5.16	5.94
4) Proposed	**6.70**	**5.59**	**5.3**	5.17	**5.69**

Table 4 shows the EER results under unmatched SNR conditions with babble noise. The proposed method also outperforms the DNN/i-vector and deep speaker. Compared with the DNN/i-vector, the proposed method improves (relative) the EER around 18.95% on average. Compared with the deep speaker, The proposed method improves (relative) the EER around 4.21% on average. In addition, we can also find that the proposed method and deep speaker are not only better than i-vector, but also less affected with SNR decreasing.

5. CONCLUSION

In this paper, we propose an architecture integrating speech separation and deep speaker for robust end-to-end text independent speaker verification. The systematic evaluation shows that our proposed method outperforms the state-of-the-art algorithms (DNN/i-vector and deep speaker). Moreover, we find that end-to-end approach is much better than DNN/i-vector in low SNRs condition. The results also indicate that the deep bottleneck feature extracted from speech separation is robust for speaker verification in noisy environment.

6. ACKNOWLEDGEMENTS

This research was partly supported by the China National Nature Science Foundation (No.61876214).

7. REFERENCES

[1] G. Heigold, I. Moreno, and S. Bengio, "End-to-end text-dependent speaker verification," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5115–5119.

[2] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech 2017*, 2017, pp. 999–1003.

[3] S. H. Ghalehjegh and R. C. Rose, "Deep bottleneck features for i-vector based text-independent speaker verification," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 555–560.

[4] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 788–798, 2011.

[5] D Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digit. Signal Process.*, pp. 19–41, 2000.

[6] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[7] G. Hinton, L. Deng, D. Yu, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, pp. 82–97, 2012.

[8] Y. Lei, N. Scheffer, L. Ferrer, and M. Mclaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1695–1699.

[9] Y. Liu, Y.M. Qian, N.X. Chen, T.F Fu, Y Zhang, and K Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, pp. 1–13, 2015.

[10] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4460–4464.

[11] L.T. Li, Y.Y. Lin, et al., "Improved deep speaker feature learning for text-dependent speaker recognition," in *Signal and Information Processing Association Summit and Conference*, 2016, pp. 426–429.

[12] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *CoRR*, vol. abs/1705.02304, 2017.

[13] F.A. Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5359–5363, 2018.

[14] D. Snyder, D Garcia-Romero, Gregory S., D. Povey, and S. Khudanpur, "x-vectors : Robust dnn embeddings for speaker recognition," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333, 2018.

[15] J. Chang and D.L. Wang, "Robust speaker recognition based on dnn/i-vectors and speech separation," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5415–5419, 2017.

[16] K. Tan and D.L. Wang, "A convolutional recurrent neural network for real-time speech enhancement," in *Proc. Interspeech 2018*, 2018, pp. 3229–3233.

[17] D.L. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1702–1726, 2018.

[18] X. Zhang and D.L. Wang, "Deep learning based binaural speech separation in reverberant environments," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1075–1084, 2017.

[19] H. Zhang, X. Zhang, and G.L. Gao, "Training supervised speech separation system to improve stoi and pesq directly," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5374–5378, 2018.

[20] Y. Wang, A. Narayanan, and D.L. Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1849–1858, 2014.

[21] Y. Xu, J. Du, L. Dai, and C. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, pp. 65–68, 2014.

[22] J. Chen and D. Wang, "Long short-term memory for speaker generalization in supervised speech separation," *Journal of the Acoustical Society of America*, vol. 141, no. 6, pp. 4705, 2017.

[23] Y. Zhang, M. Pezeshki, et al., "Towards end-to-end speech recognition with deep convolutional neuralnetworks," in *Proc. Interspeech 2016*, 2016, pp. 410–414.

[24] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[25] NIST Multimodal Information Group, "2006 nist speaker recognition evaluation training set ldc2011s09," *Philadelphia: Linguistic Data Consortium*, 2011.

[26] NIST Multimodal Information Group, "2008 nist speaker recognition evaluation training set part 1 ldc2011s05," *Philadelphia: Linguistic Data Consortium*, 2011.

[27] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," .

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