INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

The DKU Speech Activity Detection and Speaker Identification Systems for Fearless Steps Challenge Phase-02

Qingjian Lin[1,3], *Tingle Li*[1], *Ming Li*[1,2]

[1]Data Science Research Center, Duke Kunshan University, Kunshan, China
[2]School of Computer Science, Wuhan University, Wuhan, China
[3]School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China
ming.li369@dukekunshan.edu.cn

Abstract

This paper describes the systems developed by the DKU team for the Fearless Steps Challenge Phase-02 competition. For the Speech Activity Detection task, we start with the Long Short-Term Memory (LSTM) system and then apply the ResNet-LSTM improvement. Our ResNet-LSTM system reduces the DCF error by about 38% relatively in comparison with the LSTM baseline. We also discuss the system performance with additional training corpora included, and the lowest DCF of 1.406% on the Eval Set is gained with system pre-training. As for the Speaker Identification task, we employ the Deep ResNet vector system, which receives a variable-length feature sequence and directly generates speaker posteriors. The pre-training process with Voxceleb is also considered, and our best-performing system achieves the Top-5 accuracy of 92.393% on the Eval Set.

Index Terms: Fearless Steps Challenge, Speech Activity Detection, ResNet-LSTM, Speaker Identification, Deep ResNet vector

1. Introduction

Fearless Steps Challenge Phase-02 (FS02) [1, 2, 3] is the speech competition organized by UTDallas-CRSS. It aims at digitization, recovery and diarization of 19,000 hours audio data from the Apollo-11 Mission, as well as exploring meaningful information from the resource. Four tasks are released by the competition: Speech Activity Detection (SAD), Speaker Identification (SID), Speaker Diarization (SD) and Automatic Speech Recognition (ASR). In this paper, we focus on the first two tasks and present our systems.

SAD is the process of distinguishing speech regions from non-speech in audio streams [4, 5]. It serves as a fundamental front-end for massive speech signal processing technologies including ASR, keyword spotting and speech enhancement. Early works assumed that speech regions denoted speakers' voices and non-speech parts denoted the silence. Under the hypothesis, energy-based methods and zero-crossing rate were proposed [6, 7]. However, the strict definition of speech should be any sound generated by humans' vocal cords, and non-speech includes all sounds except speech. For example, laughter and coughing are categorized as speech, while background music is non-speech. To build more robust SAD systems, researchers come up with generative models like Gaussian Mixture Model (GMM) [8] and Hidden Markov Model (HMM) [9], which have stayed popular for decades. In recent years, with the development of hardware, Deep Neural Network (DNN) based models like Multi-Layer Perceptron (MLP) [10, 11] and Long Short-Term Memory (LSTM) [12, 13] successfully refresh state-of-the-art SAD performance in the literature.

Another task, SID, is the process of identifying a speaker from characteristics of voices [14, 15]. To be specific, given an utterance of variable duration, the SID system assigns it to the best matching speaker in the speaker library. SID can be either text-dependent or text-independent, and here we only discuss the text-independent case. That is, the system identifies a speaker without constraint on the speech content. Traditionally, the most typical SID system is the i-vector system [16], where the speaker-representative supervector is first extracted from GMM and then projected into the Total Variability Subspace to extract the i-vector. Then similarity measurement algorithms like cosine similarity and Probabilistic Linear Discriminant Analysis (PLDA) [17, 18] compute the scores between the newly extracted i-vector with the registered ones in the speaker library, and determine the best-matching speaker with the highest score. Alternatively, an increasing number of studies directly optimize neural networks to distinguish different speakers [19, 20]. The representative system is x-vector [21], which consists of a Time Delay Neural Network (TDNN) [22], a statistics pooling layer and a feed-forward network.

In this paper, we propose the ResNet-LSTM model for the SAD task. In comparison with the LSTM based system, the newly added ResNet front-end transforms data frames to task-relative feature mappings, and helps the LSTM back-end capture sequential information in the audio stream more easily. For the SID task, we employ Deep ResNet vector [23, 24], which is similar to x-vector in structure but replaces TDNN with deeper ResNet [25].

The rest of this paper is organized as follows. Section 2 describes our ResNet-LSTM based SAD system in detail. Section 3 introduces the Deep ResNet vector for the SID task and discusses the pre-training process in the FS02 competition. Experimental configuration and results are reported in Section 4, while conclusions are drawn in Section 5.

2. Speech Activity Detection

This section starts with the LSTM based SAD system and then introduces our ResNet-LSTM improvement. We describe the implementation of the ResNet-LSTM system in detail and list the model parameters. Besides, since FS02 competition follows open training conditions, we also discuss how to utilize additional corpora for system training.

2.1. LSTM based SAD

LSTM is popular in the SAD task due to its outstanding sequence analysis capability. Generally, sequential frame-wise features like MFCC and fbank are extracted from the audio

http://dx.doi.org/10.21437/Interspeech.2020-1915

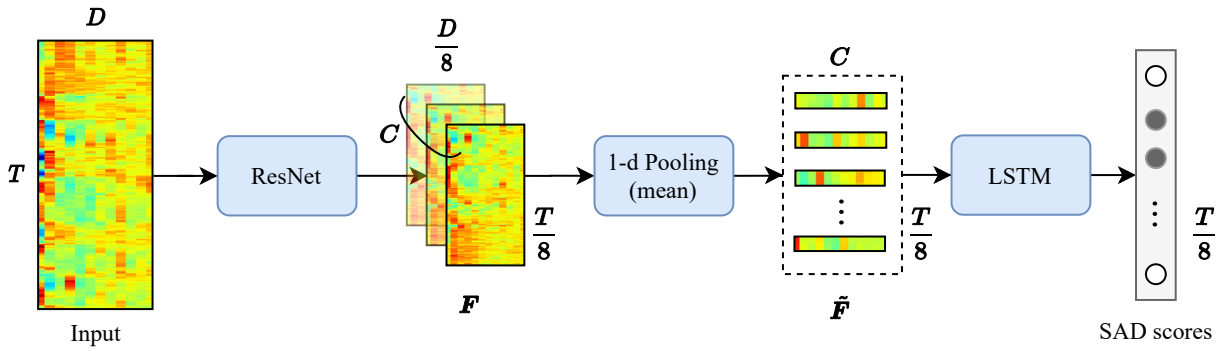

Figure 1: *The structure of ResNet-LSTM based SAD system.*

stream, fed into multiple stacked LSTM layers, and generate corresponding output scores [13]. The supervised target is the 0/1 sequence, where 1 denotes speech and 0 denotes non-speech. Compared with the pure MLP structure, LSTM is designed naturally to smooth outputs and allows short pauses in speaking to be categorized as speech.

Although the LSTM model works well in capturing sequential information, it is rarely stacked deep to extract high-level abstract feature mappings from inputs like Convolutional Neural Networks (CNN), probably due to difficulty in parallelized training [26]. Meanwhile, MFCC and fbank are fundamental features in speech signal processing and not specifically designed for the SAD task. Therefore, in more noisy and challenging scenarios, it may be hard for the system to extract essential information and generate accurate outputs.

2.2. ResNet-LSTM based SAD

To further enhance the capability of LSTM in the SAD task, we propose the ResNet-LSTM approach. As depicted in Figure 1, the neural network mainly consists of three components: a ResNet front-end, a one-dimensional (1-d) statistics pooling layer and a LSTM back-end. The LSTM back-end includes two bidirectional LSTM (Bi-LSTM) layers and a linear layer connected with the Sigmoid function.

Given sequential features of size $T \times D$, ResNet transforms the inputs to high-level feature mappings $\boldsymbol{F} \in \mathbb{R}^{C \times \frac{T}{8} \times \frac{D}{8}}$ first. T denotes the number of frames along the time axis, D denotes the feature dimension along the frequency axis, and C denotes the number of CNN channels. Then the 1-d pooling layer accumulates mean statistics over the frequency axis and generates $\tilde{\boldsymbol{F}} \in \mathbb{R}^{C \times \frac{T}{8}}$. The column vector $\tilde{\boldsymbol{F}}_t \in \mathbb{R}^{C \times 1}$ indicates the task-relative feature extracted from the $[8t, 8t+8)$ data frames. Finally, we feed the column vectors to the LSTM back-end and generate corresponding SAD scores. Detailed configuration of model parameters is listed in Table 1.

As demonstrated in the table, we employ a light-weight ResNet18 as the front-end with channel widths of residual blocks set to {16, 32, 64, 128}. After the 1-d pooling layer, each column vector $\tilde{\boldsymbol{F}}_t$ has a receptive field of 109 frames over the input features. It is worth noting that the corresponding output score s_t is only assigned to the central 8 frames, and the rest frames in the receptive field play the role of imposing additional information for system decisions. This is the main improvement in comparison with [27], where we forced sequential input features into multiple 8-frame segments and lost the additional receptive field.

Table 1: *Model parameters and output size of ResNet-LSTM.*

Layer	Parameters	Output size
Input	-	$T \times D$
ResNet	conv 3×3, 16	$16 \times T \times D$
	$\begin{bmatrix} \text{conv } 3\times3, 16 \\ \text{conv } 3\times3, 16 \end{bmatrix} \times 2$	$16 \times T \times D$
	$\begin{bmatrix} \text{conv } 3\times3, 32 \\ \text{conv } 3\times3, 32 \end{bmatrix} \times 2, /2$	$32 \times \frac{T}{2} \times \frac{D}{2}$
	$\begin{bmatrix} \text{conv } 3\times3, 64 \\ \text{conv } 3\times3, 64 \end{bmatrix} \times 2, /2$	$64 \times \frac{T}{4} \times \frac{D}{4}$
	$\begin{bmatrix} \text{conv } 3\times3, 128 \\ \text{conv } 3\times3, 128 \end{bmatrix} \times 2, /2$	$128 \times \frac{T}{8} \times \frac{D}{8}$
1-d Pooling	mean	$128 \times \frac{T}{8}$
transpose	-	$\frac{T}{8} \times 128$
Bi-LSTM	64 units per direction, 2 layers, drop=0.5	$\frac{T}{8} \times 128$
Linear	128×1, with Sigmoid	$\frac{T}{8} \times 1$

We also try deeper ResNet like ResNet34, but experimental results show very limited improvement with a wider receptive field. It indicates that the SAD decision at the t-th moment mainly relies on inputs in the range of $t \pm 0.5$ seconds.

2.3. Addtional Corpora for Training

Besides the 63-hour FS02 Train Set provided by the competition, participants are allowed to use any available data. Therefore, we also take the AMI [28] and ICSI [29] meeting corpora for system training, which sum up to 170 hours in total. Audios are resampled to 8k sample rate and mixed down to the mono channel. Moreover, non-speech regions in FS02 Train Set are truncated and mixed with the meeting audios for data augmentation.

In experiments, we consider two strategies of handling the additional meeting corpora. The first strategy combines meeting data and FS02 Train Set together for training, while the second one pre-trains systems with meeting data and then employs FS02 Train Set for model adaptation.

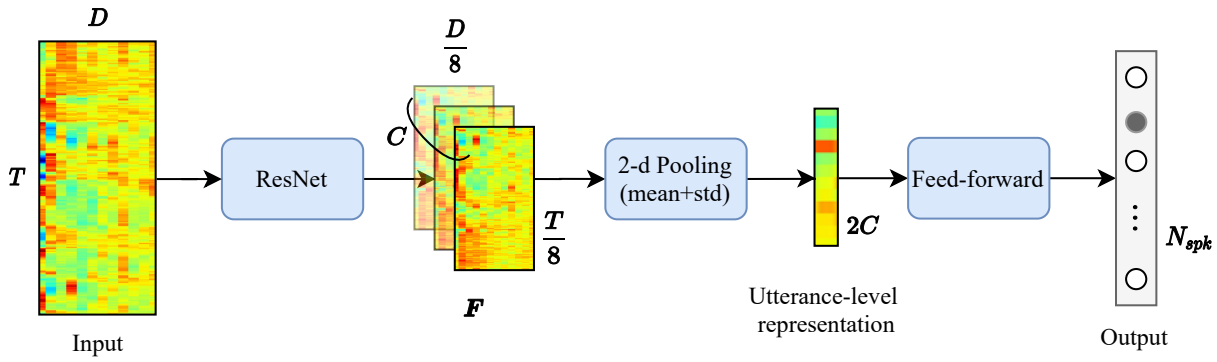

Figure 2: *The structure of the Deep ResNet vector system.*

3. Speaker Identification

In this section, we build the Deep ResNet vector system for the SID task. The system accepts a single-speaker utterance of variable duration and directly generates the speaker posteriors. We also consider additional corpora for system training.

3.1. Deep ResNet vector

As demonstrated in Figure 2, the network structure consists of three main components: a ResNet front-end, a two-dimensional (2-d) statistics pooling layer and a feed-forward network. The feed-forward network includes two stacked linear layers, connected with the Softmax function. Given sequential input features of shape $T \times D$, the ResNet front-end first converts them to frame-wise feature mappings $\boldsymbol{F} \in \mathbb{R}^{C \times \frac{T}{8} \times \frac{D}{8}}$. Then the 2-d statistics pooling layer calculates mean and standard deviation (std) statistics over the joint axis of time and frequency, generating the utterance-level representation of $2C$ dimension. Last, the feed-forward network transforms the utterance-level representation to the output with N_{spk} units. Each unit in the output layer indicates a registered speaker identity, and N_{spk} is the number of registered speakers in the library. Detailed configuration of model parameters and output size in each layer are reported in Table 2. We select ResNet34 as the front-end, with channel widths set to $\{32, 64, 128, 256\}$.

Table 2: *Model parameters and output size of Deep ResNet vector.*

Layer	Parameters	Output size
Input	-	$T \times D$
ResNet	conv 3×3, 32	$32 \times T \times D$
	$\begin{bmatrix} \text{conv } 3\times3, 32 \\ \text{conv } 3\times3, 32 \end{bmatrix} \times 3$	$32 \times T \times D$
	$\begin{bmatrix} \text{conv } 3\times3, 64 \\ \text{conv } 3\times3, 64 \end{bmatrix} \times 4, /2$	$64 \times \frac{T}{2} \times \frac{D}{2}$
	$\begin{bmatrix} \text{conv } 3\times3, 128 \\ \text{conv } 3\times3, 128 \end{bmatrix} \times 6, /2$	$128 \times \frac{T}{4} \times \frac{D}{4}$
	$\begin{bmatrix} \text{conv } 3\times3, 256 \\ \text{conv } 3\times3, 256 \end{bmatrix} \times 3, /2$	$256 \times \frac{T}{8} \times \frac{D}{8}$
2-d Pooling	mean + std	512
Linear1	256×128, dropout=0.5	128
Linear2	$128 \times N_{spk}$, with Softmax	N_{spk}

3.2. Addtional Corpora for Training

We employ the Voxceleb [30] corpus for system pre-training, which includes 7323 speakers in total. Audios are resampled to 8k sample rate. First, the Deep ResNet vector model is pre-trained with Voxceleb audios. Then we freeze parameters of the ResNet front-end and take FS02 Train Set[1] to adapt the model. Note that dimension of the output layer also reduces from 7323 to 218 in the model adaptation stage, where 218 is the number of speakers in FS02 Train Set.

4. Experiments

4.1. SAD

4.1.1. Metrics

The Detection Cost Function (DCF) is reported for the SAD task [31]. It sums up the false negative rate (fnr) and the false positive rate (fpr) by weights of 0.75 and 0.25:

$$DCF = 0.75 \times fnr + 0.25 \times fpr.$$

Short collars of 0.5s on ground-truth speech boundaries are not evaluated.

4.1.2. System Configuration

Four SAD systems are carried out in experiments:

- LSTM model trained with FS02 Train Set. It is the same as the back-end of ResNet-LSTM. Input features are sub-sampled by a factor of 8 on the time axis.
- ResNet-LSTM model trained with FS02 Train Set.
- ResNet-LSTM model trained with the mixture of additional meeting data and FS02 Train Set.
- ResNet-LSTM model pre-trained by meeting data and adapted with FS02 Train Set.

Audios are truncated into segments of 30 seconds for training, and 64-dimensional fbanks are extracted with 25 ms length and 10 ms shift. The Binary Cross Entropy (BCE) loss function computes losses between output SAD scores and ground-truth binary labels. Every time when the system finishes one epoch of

[1]FS02 Train/Dev/Eval Sets in SID sections are different from those in SAD sections.

Table 3: *DCF of the four SAD systems.*

ID	Model	Training Sets	Dev(%)	Eval(%)
1	LSTM	FS02 Train	1.828	2.44
2	ResNet-LSTM	FS02 Train	1.123	1.605
3	ResNet-LSTM	Meeting + FS02 Train (mixture)	1.060	1.751
4	ResNet-LSTM	Meeting + FS02 Train (pre-train + adapt)	**1.035**	**1.406**
-	Baseline	-	12.5	13.6

Table 4: *Top-5 accuracy of two SID systems.*

ID	Model	Training Sets	Dev(%)	Eval(%)
1	Deep ResNet vector	FS02 Train	90.789	90.751
2	Deep ResNet vector	Voxceleb + FS02 Train (pre-train + adapt)	**93.3**	**92.393**
-	Baseline	-	75.2	72.46

training, the loss on the entire FS02 Dev Set is also calculated. The Stochastic Gradient Descent (SGD) optimizer is employed, with the learning rate initialized as 0.01 and decreasing by a factor of 1/10 when the Dev loss does not improve for over 3 epochs. The training process terminates if the Dev loss gains no improvement for 10 epochs. For two-stage training where the system is pre-trained and then adapted, the same settings are applied in each stage. The checkpoint with the lowest Dev loss is selected and evaluated on FS02 Eval Set.

Systems are trained on the Pytorch deep learning framework. The hardware during the training process includes 2 GeForce GTX 1080ti GPU cards with 11 GB memory and 4 cores of Intel(R) Xeon(R) CPU E5-2630 v4.@ 2.20 GHz. For evaluation, a single CPU core is employed. The used disk storage is 3.6 GB, and the total available RAM is 8 GB.

4.1.3. Results

Results are reported in Table 3. Compared with the LSTM based SAD, our proposed ResNet-LSTM system reduces the DCF from 2.44% to 1.605% on the Eval Set. Besides, additional meeting data also brings improvement. The transfer learning strategy with pre-training and fine-tuning works better, which results in a lower DCF of 1.035% on the Dev Set, as well as 1.406% on the Eval Set in our experiments. Our best-performing system ranks third among all submissions on the leader board.

It takes 30 seconds for the ResNet-LSTM system to process a 30-minute audio with a single CPU core, and the real time factor (RTF) is 0.0167.

4.2. SID

4.2.1. Metrics

The accuracy of the Top-5 system predictions is taken as the SID metric.

4.2.2. System Configuration

Two SID systems are carried out in experiments:

- Deep ResNet vector trained with FS02 Train Set.
- Deep ResNet vector pre-trained with Voxceleb and then adapted on FS02 Train Set.

64-dimensional fbank features are extracted, with the number of frames ranging from 200 to 400 randomly for training. The Cross Entropy (CE) loss and the SGD optimizer are employed, and the Dev loss is computed after each epoch. In the pre-training stage of the second system, the learning rate is set to 0.1, 0.01, 0.001, and switches at the 25th and 40th epoch. The pre-training process terminates after 50 epochs. As for training of the first system and model adaptation of the second system, the learning rate is initialized as 0.1 and decreases by a factor of 1/10 when the Dev loss does not improve for over 3 epochs. The training process terminates if the Dev loss gains no improvement for 10 epochs. Then the checkpoint with the lowest Dev loss is selected and evaluated on F02 Eval Set.

Systems are trained with the Pytorch framework. The hardware in the training process includes 4 GeForce GTX 1080ti GPU cards with 11 GB memory and 12 cores of Intel(R) Xeon(R) CPU E5-2630 v4.@ 2.20 GHz. For evaluation, a single CPU core is employed. The used disk storage is 4.5 GB and the total available RAM is 40 GB.

4.2.3. Results

As demonstrated in Table 4, the Deep ResNet system trained with FS02 Train Set achieves a Top-5 Accuracy of 90.751% on the Eval Set. Moreover, with the pre-training process involved, the accuracy further improves to 92.393%. Both of our systems show superior performance to the official baseline developed by SincNet [32], and the second system takes the first place on the leaderboard.

The system execution time for the entire Eval Set is 44 minutes on a single CPU core, and the RTF is 0.0771.

5. Conclusions

This paper presents the DKU systems for SAD and SID tasks in the FS02 competition. In sections of SAD, we compare the structure of LSTM and ResNet-LSTM, as well as different strategies of utilizing additional training corpora.The best-performing system achieves a DCF of 1.406% on the Eval Set. In sections of SID, we employ the Deep ResNet vector to predict speaker identity of input utterances. The pre-training process with out-of-domain corpora is also explored, which further increases the Top-5 accuracy from 90.751% to 92.393%. Our best-submitted systems rank in the 3rd place for the SAD task, as well as the 1st place for the SID task.

6. Acknowledgements

This research is funded in part by the National Natural Science Foundation of China (61773413), Key Research and Development Program of Jiangsu Province (BE2019054), Six talent peaks project in Jiangsu Province (JY-074), Science and Technology Program of Guangzhou, China (202007030011, 201903010040).

7. References

[1] J. H. Hansen, A. Sangwan, A. Joglekar, A. E. Bulut, L. Kaushik, and C. Yu, "Fearless steps: Apollo-11 corpus advancements for speech technologies from earth to the moon," in *Interspeech*, 2018, pp. 2758–2762.

[2] J. H. Hansen, A. Joglekar, M. C. Shekhar, V. Kothapally, C. Yu, L. Kaushik, and A. Sangwan, "The 2019 Inaugural Fearless Steps Challenge: A Giant Leap for Naturalistic Audio," in *Interspeech*, 2019, pp. 1851–1855.

[3] A. Joglekar, J. H. Hansen, M. C. Shekhar, and A. Sangwan, "Fearless steps challenge (fs-2): Supervised learning with massive naturalistic apollo data," in *Interspeech*, 2020.

[4] A. Ziaei, L. Kaushik, A. Sangwan, J. H. Hansen, and D. W. Oard, "Speech activity detection for nasa apollo space missions: Challenges and solutions," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[5] T. Drugman, Y. Stylianou, Y. Kida, and M. Akamine, "Voice activity detection: Merging source and filter-based information," *IEEE Signal Processing Letters*, vol. 23, no. 2, pp. 252–256, 2015.

[6] K. R. Borisagar, D. G. Kamdar, B. S. Sedani, and G. Kulkarni, "Speech enhancement in noisy environment using voice activity detection and wavelet thresholding," in *2010 IEEE International Conference on Computational Intelligence and Computing Research*. IEEE, 2010, pp. 1–5.

[7] M. Jalil, F. A. Butt, and A. Malik, "Short-time energy, magnitude, zero crossing rate and autocorrelation measurement for discriminating voiced and unvoiced segments of speech signals," in *2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering*. IEEE, 2013, pp. 208–212.

[8] A. Tsiartas, T. Chaspari, N. Katsamanis, P. K. Ghosh, M. Li, M. Van Segbroeck, A. Potamianos, and S. Narayanan, "Multi-band long-term signal variability features for robust voice activity detection." in *Interspeech*, 2013, pp. 718–722.

[9] C. Wooters, J. Fung, B. Peskin, and X. Anguera, "Towards robust speaker segmentation: The icsi-sri fall 2004 diarization system," in *Fall 2004 Rich Transcription Workshop*, 2004.

[10] S. Ganapathy, P. Rajan, and H. Hermansky, "Multi-layer perceptron based speech activity detection for speaker verification," in *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2011, pp. 321–324.

[11] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Veselỳ, and P. Matějka, "Developing a speech activity detection system for the darpa rats program," in *Thirteenth annual conference of the international speech communication association*, 2012.

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

[13] F. Eyben, F. Weninger, S. Squartini, and B. Schuller, "Real-life voice activity detection with lstm recurrent neural networks and an application to hollywood movies," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 483–487.

[14] S. S. Tirumala, S. R. Shahamiri, A. S. Garhwal, and R. Wang, "Speaker identification features extraction methods: A systematic review," *Expert Systems with Applications*, vol. 90, pp. 250–271, 2017.

[15] R. Togneri and D. Pullella, "An overview of speaker identification: Accuracy and robustness issues," *IEEE circuits and systems magazine*, vol. 11, no. 2, pp. 23–61, 2011.

[16] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Interspeech*, 2011, pp. 249–252.

[17] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[18] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "Plda for speaker verification with utterances of arbitrary duration," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7649–7653.

[19] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2014, pp. 4052–4056.

[20] P. Kenny, T. Stafylakis, P. Ouellet, V. Gupta, and M. J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition." in *Odyssey 2014 The Speaker and Language Recognition Workshop*, vol. 2014, 2014, pp. 293–298.

[21] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5329–5333.

[22] K. J. Lang, A. H. Waibel, and G. E. Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural networks*, vol. 3, no. 1, pp. 23–43, 1990.

[23] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 74–81.

[24] W. Cai, J. Chen, J. Zhang, and M. Li, "On-the-fly data loader and utterance-level aggregation for speaker and language recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1038–1051, 2020.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[27] Q. Lin, W. Cai, L. Yang, J. Wang, J. Zhang, and M. Li, "Dihard ii is still hard: Experimental results and discussions from the dku-lenovo team," in *Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020.

[28] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The ami meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*. Springer, 2005, pp. 28–39.

[29] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke *et al.*, "The icsi meeting corpus," in *2003 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2003.

[30] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Interspeech*, 2017, pp. 2616–2620.

[31] F. Byers and O. Sadjadi, *2017 Pilot Open Speech Analytic Technologies Evaluation (2017 NIST Pilot OpenSAT): Post Evaluation Summary*. US Department of Commerce, National Institute of Standards and Technology, 2019.

[32] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *2018 IEEE Spoken Language Technology Workshop*, 2018, pp. 1021–1028.