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

The JD AI Speaker Verification System for the FFSVC 2020 Challenge

Ying Tong, Wei Xue, Shanluo Huang, Lu Fan, Chao Zhang, Guohong Ding, Xiaodong He

JD AI Research

{tongying,xuewei27,huangshanluo,fanlu,chao.zhang,dingguohong,xiaodong.he}@jd.com

Abstract

This paper presents the development of our systems for the Interspeech 2020 Far-Field Speaker Verification Challenge (FFSVC). Our focus is the task 2 of the challenge, which is to perform far-field text-independent speaker verification using a single microphone array. The FFSVC training set provided by the challenge is augmented by pre-processing the far-field data with both beamforming, voice channel switching, and a combination of weighted prediction error (WPE) and beamforming. Two open-access corpora, CHData in Mandarin and VoxCeleb2 in English, are augmented using multiple methods and mixed with the augmented FFSVC data to form the final training data. Four different model structures are used to model speaker characteristics: ResNet, extended time-delay neural network (ETDNN), Transformer, and factorized TDNN (FTDNN), whose output values are pooled across time using the self-attentive structure, the statistic pooling structure, and the GVLAD structure. The final results are derived by fusing the adaptively normalized scores of the four systems with a two-stage fusion method, which achieves a minimum of the detection cost function (minDCF) of 0.3407 and an equal error rate (EER) of 2.67% on the development set of the challenge.

Index Terms: speaker verification, deep neural network, data augmentation, score normalization

1. Introduction

Over the past few years, due to the rapid development of deep learning, the performance of near-field speaker recognition systems has achieved substantial improvements [1–3]. However, when the microphone is located far from the speaker, the quality of the speech signals captured are more likely to be affected by energy decaying, reverberation, and environmental noise, which makes the speaker recognition problem more challenging [4]. Recently, there is an increasing interest in far-field speech processing, due to the prevalence of the emerging speech applications in smart-home devices, robotics, and audio surveillance. Microphone arrays are often used to capture the far-field speech signals, which can provide additional spatial information in the derived acoustic features. Though much efforts have been made for far-field automatic speech recognition(ASR) [5, 6], fewer studies have focused on speaker recognition by far.

The Interspeech 2020 Far-Field Speaker Verification Challenge (FFSVC) is launched to facilitate the study on both far-field text-dependent and text-independent speaker verification [7–10] problems. In this paper, we describe our systems and the experimental results on FFSVC task 2, the text-independent speaker verification with a single microphone array. It is an open-track task since external open-access datasets are allowed to be used along with the officially-released 1,100-hour far-field training data (denoted as FFSVC20). Since the external datasets are recorded with different acoustic environments, namely source-sensor distances, transmission channels, and microphone frequency responses, how to use them to improve the far-field speaker recognition performance on the in-domain FFSVC20 data is a problem. We comprehensively described the use of near-field to far-field transformation (near-to-far) and its reverse based on signal processing, as well as different data augmentation methods related to additive/convolutional noises and room impulse responses (RIRs). Different DNN structures and fusion methods are also investigated [11]. Both far-to-near (dereverberation) and near-to-far (reverberation) transformations are used for FFSVC and the external datasets to capture better channel-invariant for speaker characteristics. More specifically, regarding the FFSVC dataset that includes the same utterance recorded by microphones at different distances, signal-processing based beamforming method [12] is used to reduce the noise and reverberation using the spatial information. Moreover, the weighted prediction error (WPE) [13] is adapted to further reduce the reverberation using the temporal correlation. For the external datasets, data augmentation by transforming the near-field data to far-field is used to increase channel and acoustic environment information for speaker characteristics to fit the characteristics of FFSVC. The near-to-far data augmentation is implemented using a linear convolution using the RIRs estimated for each pair of near-field and far-field microphones in the FFSVC dataset.

All of our systems are deep-learning-based. Four different structures are used as encoder, namely ResNet [14], extend time-delay neural network (ETDNN) [15, 16] , Transformer [17], and factorized TDNN (FTDNN) encoder. For each type of model, the output values are pooled across time using the self-attentive [18–20] or the statistic pooling structure [2].

The angular softmax function is introduced in this work to increase the discrimination between the speakers and decrease the distance of the intra-speakers. Back-end scoring is performed using probabilistic linear discriminant analysis (PLDA) [21] and cosine similarity [22]. Adaptive score normalization [23] is used to increase the robustness against different channels, and the normalized scores from different systems are fused using BOSARIS toolkit [24] in the end. The remainder of the paper is organized as follows. Section 2 describes the data preparation pipeline. Section 3 presents the structures of the systems. Experimental results are presented in Section 4, followed by conclusions.

2. Data Preparations

This section describes our data preparation pipelines.

2.1. Pipeline for FFSVC20

The FFSVC20 dataset was collected in multiple scenarios. The training set includes 120 speakers with a total number of 1,100-hour speech, and the development set consists of data from 35

The authors would like to thank Dr. Shuai Wang and Dr. Yi Liu for useful discussions and suggestions.

http://dx.doi.org/10.21437/Interspeech.2020-3062

speakers. FFSVC20 uses a close-talking microphone, an iPhone at 25 cm distance and three randomly selected 4-channel microphone arrays in the training and development sets.

The target of task 2 of FFSVC is to determine whether the far-field speech signal captured by a microphone array belongs to a certain speaker according to the enrollment utterances recorded by the near-field devices. The following four data augmentation methods are used, which can help the DNN speaker classifier to capture better channel- and acoustic-environment-invariant features for speaker discrimination.

2.1.1. Dereverberation

The recordings of the FFSVC20 dataset were obtained under complex and varied acoustic environments. In such conditions, reverberation is a core factor that impacts the recognition accuracy. To alleviate that, The NARA-WPE toolbox [25] based on the weighted prediction error (WPE) method is applied to the far-field speech to reduce the effect of reverberation.

2.1.2. Beamforming

The quality of far-field speech is degraded as a result of the attenuated energy of the direct-path speech signal. Both of the direct-to-reverberation ratio (DRR) and the signal-to-noise ratio (SNR), which jointly describe the target speech dominance in the captured far-field signal, are decreased. With multiple microphones, the beamformer performs spatial filtering by steering a beam towards the speaker's direction and suppresses the background noise and the reverberation from other directions.

It has been shown that using beamforming to pre-process the data can improve the robustness of the speaker verification system in complex acoustic environments [26]. In this paper, beamforming is performed for the far-field multichannel signals from the FFSVC20 dataset, and the resulting utterances are used to augment the datasets. In addition, due to the existence of sidelobes of the beamformer, the reverberation and noise are "suppressed" rather than "removed" in the data generated by beamforming, which can be seen as a reasonable interpolation between the near-field and far-field signals. Such an interpolation smoothed the feature space and can improve the DNN model performance. The BeamformIt toolbox [12] is used to perform beamforming, which takes an arbitrary number of input channels without any prior information and computes an output by filter-and-sum beamforming.

Moreover, the far-field signals pre-processed by WPE and beamforming (denoted as WPEBF) are used in the PLDA training stage or the scoring stage.

2.1.3. Voice channel switching

The voice channel switching method proposed in [27] can simulate the rotation relationship to capture more spatial information. Here we extend this idea to combine signals from different distances with different channels for the beamforming process. Such an operation could help to augment data with diversity acoustic environment while not change the spatial information of the original data set. In this paper, voice channel switching is applied to the FFSVC20 far-field data training set, and the augmented versions are used in the PLDA training stage.

2.1.4. PyRIR

Room impulse response (RIR) is the transfer function between the sound source and the receiver devices, which carry the acoustic characteristics in the process of sound propagation. Although the FFSVC20 provides multi-channel far-field signals, the relative position of the microphone device and the room size is fixed. The Pyroomacoustics toolkit [28] provide the function for quickly construct various simulate scenarios, which can help to increase channel information. The FFSVC20 near-field data signals are convolving with the simulated RIRs, and the resulting utterances are used to augmented the training sets.

2.2. Pipeline for external open-access datasets

In this paper, two open-access speaker recognition datasets, CHData [1] and VoxCeleb2 [29], are used as the external open-access datasets to construct the systems. For CHData, the subsets SLR$\{18, 33, 47, 62, 68, 85\}$ are selected to use, which consists of a total number of 2897 hours of speech 5126 speakers. For VoxCeleb2, a total number of 5994 speakers are used. We have noticed that the SLR85 subset of CHData is recorded similarly to FFSVC20, and the other CHData subsets are the near-field data. Together with the PyRIR method described in Section 2.1.4, another two data augmentation methods, the SIAug and SREAug, are also applied to the open-access datasets.

2.2.1. SIAug

Since the data included in the open-access corpora mostly consist of near-field speech signals, it is crucial to simulate a far-field environment to matched how the FFSVC20 dataset was created. For each pair of the near-field/far-field signals from FFSVC20, a large collection of RIRs are estimated by performing system identification (SI) [30] from the near-field source signal to the signal captured by the far-field microphone array. All near-field training data are augmented by convolving the near-field signals with a randomly selected RIR.

2.2.2. SREAug

The SREAug pipeline from the x-vector-based speaker recognition system in the Kaldi SRE16 recipe [31] is used to increase the diversity of noise interferences and RIRs in the dataset. The SREAug contains the following steps:

a) Mixing with babel, music and noise signals from the MUSAN corpus [32];

b) Convolving with the RIRs from the AIR dataset [32].

2.3. Data pre-processing

The full training set consists of the official FFSVC20 dataset, the two open-access datasets, and their augmented versions. Three kinds of acoustic features including 60-dimension (-dim) log-Mel filter-banks (FBK) +Pitch (FBKP), 80-dim FBK+Pitch, and 30-dim PLP+Pitch (PLPP) are employed in this task. Audios are resampled to 16 kHz, and all the features are extracted from the raw signals with 25 ms frame length and 10 ms overlap. The energy-based voice activity detection (VAD) from the Kaldi SRE16 recipe is used to select the speech period. Then the features are processed through local cepstral mean normalization over a 3-second sliding window before fed into the deep speaker network.

3. Model Architectures

In this section, we introduce four types of DNN architectures and the score normalization used by our system.

[1] https://openslr.org/resources.php

Table 1: *Detailed specifications of the different the ResNet-based systems.*

Layer name	**ResNet-18**	**ResNet-34**	**ResNet-50**
Input	-	-	-
Conv2D-1	3×3, Stride 1	3×3, Stride 1	3×3, Stride 1
ResNetBlock-1	$\begin{bmatrix} 3\times 3 & 64 \\ 3\times 3 & 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3 & 64 \\ 3\times 3 & 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1 & 64 \\ 3\times 3 & 64 \\ 1\times 1 & 256 \end{bmatrix} \times 3$
ResNetBlock-2	$\begin{bmatrix} 3\times 3 & 128 \\ 3\times 3 & 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3 & 128 \\ 3\times 3 & 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times 1 & 128 \\ 3\times 3 & 128 \\ 1\times 1 & 512 \end{bmatrix} \times 4$
ResNetBlock-3	$\begin{bmatrix} 3\times 3 & 256 \\ 3\times 3 & 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3 & 256 \\ 3\times 3 & 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times 1 & 256 \\ 3\times 3 & 256 \\ 1\times 1 & 1024 \end{bmatrix} \times 6$
ResNetBlock-4	$\begin{bmatrix} 3\times 3 & 512 \\ 3\times 3 & 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3 & 512 \\ 3\times 3 & 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1 & 512 \\ 3\times 3 & 512 \\ 1\times 1 & 2048 \end{bmatrix} \times 3$
Conv2D	1× K, Stride 1		
Fully Connected Layer	512× 512 (Input × output)		
Fully Connected Layer	512× 1500 (Input × output)		
Self-attentive Pooling Layer	1500× 3000 (Input × output)		
Fully Connected Layer	3000× 512 (Input × output)		
Fully Connected Layer	512× 512 (Input × output)		
ArcSoftmax	512× N (Input × output)		

Table 2: *Detailed specifications of the FTDNN-based system.*

Num	Layer	Context Factor 1	Context Factor 2	Skip Conn. from Layer
1	TDNN	t-2:t+2		
2	FTDNN	t-2,t	t,t+2	
3	FTDNN	t	t	
4	FTDNN	t-3,t	t,t+3	
5	FTDNN	t	t	3
6	FTDNN	t-3,t	t,t+3	
7	FTDNN	t-3,t	t,t+3	2,4
8	FTDNN	t-3,t	t,t+3	4,6,8
9	FTDNN	t	t	
10	None/Lstm	t	t	
11	Dense	t	t	
12	Pooling	Full Seq.		
13	Dense	[0, T]		
14	Dense	[0, T]		
15	Softmax	[0, T]		

3.1. DNN-based systems

All of our systems are deep speaker embeddings-based, which accept variable-length segments and produce an utterance-level score. The ETDNN, ResNet, Transformer, and FTDNN based systems are developed, and the main differences of these systems are in the encoders prior to the pooling layers.

3.1.1. ETDNN-based systems

We use a bigger network with more neurons in extended-TDNN layers. The detailed description of the network is summarized in [16]. The first 10 layers of the x-vector system operate on the frame level, with a small temporal context window centered at the current frame t, followed by a statistic pooling layer. Then the segment-level statistics are concatenated and passed through the segment-level layers.

3.1.2. ResNet-based systems

Table 1 summarizes the adopted ResNet-based network architecture. Similar to the ETDNN based system, the ResNet-based system differences among ResNet-18/ResNet-34/ResNet-50 are in the depth and structure of the residual layer. Unlike the original ResNet-based architecture, we process the feature from the residual layer 4 via a convolution layer and two fully-connected layers to prevent reducing the time resolution. The ArcSoftmax loss [34] was utilized to further increase the distance between the speakers while retaining a small intra-speaker distance. Besides the ResNet-based network shown in Table 1, we also test the structure of Thin-ResNet-34 with the GVLAD pooling layer proposed in [35].

3.1.3. Transformer-based systems

The Transformer-based system is a combination of ResNet-Block layer and identical layer, followed by a statistic pooling layer. The identical layer is composed of a multi-head self-attention layer and a position-wise fully-connected layer.

3.1.4. FTDNN-based systems

Unlike the traditional x-vector architecture, the traditional TDNN layers are replaced by a factorized TDNN (FTDNN) with a skip connection. The detailed description of the FTDNN x-vector architecture is summarized in Table 2. Three kinds of FTDNN models are compared in this task.

- **FTDNN-LSTM1**: This system is shown in Table 2 with statistic pooling. The output size of each layer is 512, and the inner size of the FTDNN layer is 128. Two LSTM layers with 512-dim cell, 256-dim recurrent and non-recurrent projection units, are added after the FTDNN layer.

Table 3: *Detailed results of our systems, where "Cosine" refers to the Cosine similarity.*

System	PLDA	Cosine	PLDA (AS-norm)	Cosine (AS-norm)	Cosine+PLDA
	minDCF/%EER	minDCF/%EER	minDCF/%EER	minDCF/%EER	minDCF/%EER
FFSVC Baseline [33]	-	-	-	-	0.5800 / 5.83
Res18-att-FBKP60 (no aug)	0.9047 / 9.28	0.8992 / 9.01	-	-	-
Res18-att-FBKP60	0.7654 / 5.17	0.5735 / 4.93	0.5732 / 4.97	0.4948 / 3.95	0.4575 / 3.56
Res34-att-FBKP60	0.7382 / 4.87	0.5251 / 4.07	0.5338 / 4.44	0.4664 / 3.44	0.4290 / 3.10
Res34-stat-FBKP60	0.7304 / 4.85	0.5049 / 4.01	0.5273 / 4.49	0.4487 / 3.42	0.4131 / 3.22
Res50-att-FBKP60	0.7336 / 5.38	0.6374 / 5.58	0.5851 / 5.03	0.5580 / 4.50	0.5027 / 4.03
ThinRes34-GVLAD-FBKP60	0.8055 / 6.66	0.6602 / 5.72	0.6906 / 6.19	0.6068 / 5.09	0.5570 / 4.71
Transformer-stat-FBKP60	0.7462 / 6.70	0.6150 / 5.99	0.6215 / 6.01	0.5511 / 5.02	0.5130 / 4.86
ETDNN-stat-FBKP60	0.7882 / 5.81	0.5572 / 5.13	0.6049 / 5.73	0.5550 / 4.77	0.4975 / 4.44
EFTDNN-att-FBKP60	0.7495 / 6.38	0.5800 / 5.07	0.6591 / 6.05	0.5061 / 4.20	0.4849 / 3.97
FTDNN-LSTM1-sta-FBKP60	0.8265 / 5.50	0.5721 / 5.07	0.5927 / 5.46	0.5442 / 4.56	0.4910 / 4.26
FTDNN-LSTM2-sta-FBKP60	0.7776 / 5.81	0.5572 / 5.13	0.5513 / 5.05	0.4895 / 3.75	0.4482 / 3.63
Res34-att-FBKP80	0.6755 / 4.74	0.5223 / 4.46	0.5357 / 4.50	0.4601 / 3.60	0.4293 / 3.18
ETDNN-stat-FBKP80	0.7993 / 5.82	0.5571 / 5.16	0.6102 / 5.01	0.5510 / 4.57	0.4988 / 4.49
ETDNN-stat-PLPP30	0.8214 / 6.25	0.6475 / 5.22	0.6449 / 6.11	0.5706 / 4.78	0.5394 / 4.56
Res34-att-FBKP60-WPEBF	0.7425 / 5.13	0.5706 / 4.89	0.5241 / 4.27	0.4419 / 3.32	0.4142 / 2.99
Res34-stat-FBKP60-WPEBF	0.7195 / 4.83	0.4791 / 3.95	0.5315 / 4.42	0.4349 / 3.32	0.4080 / 3.08
Fusion	-	-	-	-	**0.3407 / 2.67**

- **FTDNN-LSTM2**: This system has a similar structure with the FTDNN-LSTM1 system, except that the output size of each layer in the frame layer is 1024, and the inner size of the FTDNN layer is 256.
- **EFTDNN**: The extended FTDNN introduced in [36] is a combination of ETDNN and FTDNN structure. Angular softmax loss is used in this system.

3.2. Adaptive score normalization

Score normalization is used to convert the network outputs from different models to a unified range, such that the scores are calibrated and a more reliable threshold can be applied. In the adaptive score normalization, only top X closest files are selected as the cohort to compute mean and variance for normalization.

4. Experimental Results

In this section, we report the results of the four deep speaker embedding-based systems as well as their fusions on the development data. Results are reported in terms of the primary evaluation metric used by FFSVC, which are the minDCF with $P_{target} = 0.01$ and EER. During testing, the scores of different channels and their augmentations in the same microphone array are equally weighted. The results shown in Table 3 are processed by the Adaptive S-norm method.

The performance of the ETDNN system with different input features is shown in Table 3. The system with FBKP60 features achieves slightly better performance compared to the system with FBKP80 features, and achieves 0.0419 and 0.12% absolute improvement on minDCF and EER, respectively, compared to the system with PLPP30 features.

The performance of different models is shown in Table 3. We evaluate the ResNet-based systems, ETDNN-based systems, the Transformer-based system, and FTDNN-based systems using the FBKP60 features. We have two major observations from the results. First, the ResNet-based systems outperform other systems. Second, FTDNN-LSTM based systems outperform the ETDNN-based systems. The best performance among the single systems is obtained using a combination of the Res34-stat-FBKP60 system and the cosine similarity scoring method, yielding a minDCF of 0.4487 and an EER of 3.42%.

Next, different data augmentation methods are compared. Comparing to the system without any augmentation policies, the combination of adding the augment data yield substantial improvements. It is clear that using the WPEBF augmentation policy could help to reduce the EER in all cases, and reduce the minDCF using the Cosine similarity scoring back-end.

The scores on the right side of Table 3 are obtained using the first-stage of the fusion method, which computes the PLDA and cosine scores by applying a simple weighted average within the same front-end training model. Comparing to the best performances of Res18-att-FBKP60, Res34-att-FBKP60, Res34-stat-FBKP60, Res50-att-FBKP60 and ThinRes34-GVLAD-FBKP60 based system, fusion scores of these systems are improved by up to 0.0373, 0.0374, 0.0356, 0.0553 and 0.0498 in the absolute value on minDCF, respectively.

The results processed by first-stage are then fused using the BOSARIS toolkit. To select the combination systems with the best performance, the greedy fusion method is adopted in this paper. The best performance of the fusion systems achieves a minDCF of 0.3407 and EER of 2.67% on the development set.

5. Conclusions

This paper describes the development of the JD AI speaker verification system for task 2 of FFSVC 2020. Various augmentation methods are used to increase the diversity of the data, which yields substantial improvements in terms of both minDCF and EER. The ResNet-based, ETDNN-based, Transformer-based, and TDNNF-based systems are investigated in this paper. By jointly using a score normalization method over 9 different systems and a two-stage score fusion method to combine their output scores, promising results are obtained with a minDCF of 0.3407 and an EER of 2.67% on the official development set.

6. References

[1] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, 2014, pp. 4052–4056.

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

[3] Z. Wu, P. L. De Leon, C. Demiroglu, A. Khodabakhsh, S. King, Z. Ling, D. Saito, B. Stewart, T. Toda, M. Wester, and J. Yamagishi, "Anti-spoofing for text-independent speaker verification: An initial database, comparison of countermeasures, and human performance," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 768–783, 2016.

[4] S. Novoselov, A. Gusev, A. Ivanov, T. Pekhovsky, A. Shulipa, G. Lavrentyeva, V. Volokhov, and A. Kozlov, "STC speaker recognition systems for the VOiCES from a distance challenge," in *arXiv preprint arXiv:1904.06093*, 2019.

[5] T. N. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, "Multichannel signal processing with deep neural networks for automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 965–979, 2017.

[6] Z. Tang, L. Chen, B. Wu, D. Yu, and D. Manocha, "Improving reverberant speech training using diffuse acoustic simulation," in *Proc. ICASSP*, 2020, pp. 6969–6973.

[7] X. Qin, M. Li, H. Bu, R. K. Das, W. Rao, S. Narayanan, and H. Li, "The FFSVC 2020 evaluation plan," *arXiv preprint arXiv:2002.00387*, 2020.

[8] X. Qin, D. Cai, and M. Li, "Far-field end-to-end text-dependent speaker verification based on mixed training data with transfer learning and enrollment data augmentation," *Proc. Interspeech*, pp. 4045–4049, 2019.

[9] S. Wang, J. Rohdin, L. Burget, O. Plchot, Y. Qian, and K. Yu, "On the usage of phonetic information for text-independent speaker embedding extraction," in *Proc. Interspeech*, 2019, pp. 1148–1152.

[10] L. You, W. Guo, L. Dai, and J. Du, "Multi-task learning with high-order statistics for x-vector based text-independent speaker verification," in *Proc. Interspeech*, 2019, pp. 1158–1162.

[11] A. Kanagasundaram, S. Sridharan, S. Ganapathy, P. Singh, and C. B. Fookes, "A study of x-vector based speaker recognition on short utterances," in *Proc. Interspeech*, 2019, pp. 2943–2947.

[12] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[13] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[15] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *Proc. ICASSP*, 2019, pp. 5796–5800.

[16] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "BUT system description to VoxCeleb speaker recognition challenge 2019," in *The VoxSRC Workhsop 2019*, 2019.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5998–6008.

[18] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification." in *Proc. Interspeech*, 2018, pp. 3573–3577.

[19] T. K. Koji Okabe and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech*, 2018, pp. 2252–2256.

[20] P. Safari and J. Hernando, "Self multi-head attention for speaker recognition," in *Proc. Interspeech*, 2019.

[21] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Odyssey*, 2010, p. 14.

[22] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Proc. Interspeech*, 2009, pp. 1559–1562.

[23] P. Matejka, O. Novotnỳ, O. Plchot, L. Burget, M. D. Sánchez, and J. Cernockỳ, "Analysis of score normalization in multilingual speaker recognition." in *Proc. Interspeech*, 2017, pp. 1567–1571.

[24] N. Brümmer and E. De Villiers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF," *arXiv preprint arXiv:1304.2865*, 2013.

[25] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing," in *13. ITG Fachtagung Sprachkommunikation (ITG 2018)*, Oct 2018, pp. 1–5.

[26] L. Mošner, P. Matějka, O. Novotnỳ, and J. H. Černockỳ, "Dereverberation and beamforming in far-field speaker recognition," in *Proc. ICASSP*, 2018, pp. 5254–5258.

[27] Q. Wang, H. Wu, Z. Jing, F. Ma, Y. Fang, Y. Wang, T. Chen, J. Pan, J. Du, and C.-H. Lee, "The USTC-iFlytek system for sound event localization and detection of DCASE2020 challenge," DCASE2020 Challenge, Tech. Rep., July 2020.

[28] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *Proc. ICASSP*, 2018, pp. 351–355.

[29] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.

[30] Y. Huang and J. Benesty, "A class of frequency-domain adaptive approaches to blind multichannel identification," *IEEE Transactions on signal processing*, vol. 51, no. 1, pp. 11–24, 2003.

[31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[32] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. ICASSP*, 2017, pp. 5220–5224.

[33] X. Qin, M. Li, H. Bu, W. Rao, R. K. Das, S. Narayanan, and H. Li, "The Interspeech 2020 far-field speaker verification challenge," *http://2020.ffsvc.org/BaselinePaper*, 2020.

[34] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proc. CVPR*, 2019, pp. 4690–4699.

[35] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *Proc. ICASSP*, 2019, pp. 5791–5795.

[36] Y. Liu, T. Liang, C. Xu, X. Zhang, X. Chen, W.-Q. Zhang, L. He, D. Song, R. Li, Y. Wu, P. Ouyang, and S. Yin, "THUEE system description for NIST 2019 SRE CTS challenge," *arXiv preprint arXiv:1912.11585*, 2019.