



Bidirectional Multiscale Feature Aggregation for Speaker Verification

Jiajun Qi, Wu Guo, Bin Gu

National Engineering Laboratory for Speech and Language Information Processing,
University of Science and Technology of China, Hefei, China

jiajun97@mail.ustc.edu.cn, guowu@ustc.edu.cn, bin2801@mail.ustc.edu.cn

Abstract

In this paper, we propose a novel bidirectional multiscale feature aggregation (BMFA) network with attentional fusion modules for text-independent speaker verification. The feature maps from different stages of the backbone network are iteratively combined and refined in both a bottom-up and top-down manner. Furthermore, instead of simple concatenation or elementwise addition of feature maps from different stages, an attentional fusion module is designed to compute the fusion weights. Experiments are conducted on the NIST SRE16 and VoxCeleb1 datasets. The experimental results demonstrate the effectiveness of the bidirectional aggregation strategy and show that the proposed attentional fusion module can further improve the performance.

Index Terms: speaker verification, deep speaker embedding, multiscale aggregation, attentional fusion

1. Introduction

Speaker verification (SV) is used to verify a person's claimed identity from their voice characteristics. During the last decade, the i-vector [1] algorithm combined with a probabilistic linear discriminant analysis (PLDA) [2] backend for similarity scoring has become very popular due to its good performance and ability to compensate for within-speaker variations.

In recent years, this paradigm has shifted to employ deep neural networks (DNNs) to extract speaker embeddings, called the d-vector [3] or x-vector [4]. DNN-based speaker embedding extraction outperforms conventional i-vectors and achieves state-of-the-art performance in many cases, especially in conditions of short-duration SV. These embedding networks typically consist of three main components: (1) several frame-level layers to obtain high-level feature representations, (2) a pooling layer that aggregates the frame-level representations across the temporal dimension and (3) a bottleneck layer that projects the pooled vector into a low-dimensional speaker embedding.

The frame-level embeddings are usually modeled by time-delay neural networks (TDNNs) [5, 6, 7], long short-term memory (LSTM) networks [8], or convolutional neural networks (CNNs) [9, 10]. Similar to the image classification field, the deep residual network (ResNet) [11] has also been widely adopted as the backbone architecture of embedding networks in SV systems [12, 13, 14] due to its good performance. ResNet usually comprises an input convolutional layer and 4 stages, where each stage represents a group of residual convolutional layers that output feature maps of the same scale.

Generally, the DNNs of the embedding extractor contain many layers. To exploit hierarchical time-frequency context information from different layers, multiscale feature aggregation (MFA) has been developed [7, 9, 14, 15, 16]. Gao et al. [14] proposed aggregating multistage feature maps with different resolutions using concatenation. Seo et al. [15] proposed con-

catenating pooled embeddings from each stage to build shortcut connections from low- and middle-layer features to the final speaker embeddings. Similar to [15], Hajavi et al. [16] used weighted summation instead of concatenation on pooled embeddings from multiple stages for aggregation. These MFA architectures extract features from different layers and aggregate them with elementwise addition or concatenation operations. More recently, a progressive MFA method [17] was introduced to create multiscale features that have high-level speaker information at all layers by applying feature pyramid module (FPM), which consists of a top-down pathway and lateral connection.

Existing MFA methods have achieved outstanding performance on SV tasks; however, there is still room for further improvement of MFA. First, the abovementioned works aggregate the multistage features only through a unidirectional pathway, and this setup may limit the feature interaction of lower and higher layers. Second, concatenation and elementwise addition are usually employed in feature aggregation steps, and these simple operations have limited abilities to extract relevant information from different features due to their fixed fusion weights.

In this work, we propose a novel bidirectional multiscale feature aggregation (BMFA) network with attentional fusion modules (AFM) for speaker verification. For the proposed method, features of different stages are first extracted from the ResNet backbone. Then, a bidirectional feature aggregation strategy, which consists of top-down and bottom-up pathways, is used to progressively fuse these features. Additionally, inspired by [18], an attentional fusion module is employed to learn better fusion weights. We evaluate our experiments on the NIST SRE16 [19] and VoxCeleb1 [20] datasets. The experimental results show that the proposed BMFA can achieve better performance than the existing MFA strategy and obtain further improvement when combined with the proposed attentional fusion module.

The rest of the paper is organized as follows. Section 2 reviews the existing architecture of MFA networks. Section 3 describes the proposed bidirectional model and attentional fusion module in detail. Then, the experimental setup is presented in section 4. The results and the analysis are presented in section 5. Finally, the conclusions are given in section 6.

2. Multiscale Feature Aggregation

In this section, we briefly review different MFA methods in SV field. The architecture of conventional MFA methods [14, 15, 16] is illustrated in Fig. 1 (a). Specifically, the feature maps of multiple scales are extracted from different stages in the ResNet-based network and then aggregated simultaneously either before or after temporal pooling. Concatenation or weighted addition is usually employed as the fusion method.

A progressive top-down MFA approach using FPM is proposed in [17], as shown in Fig. 1 (b). The FPMs iteratively

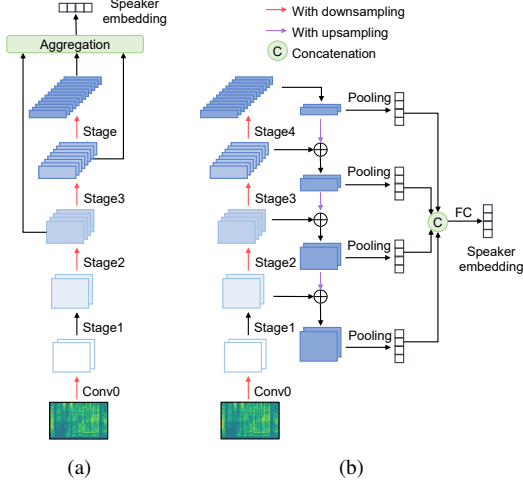


Figure 1: Illustration of different multiscale aggregation architectures in SV. (a) Feature aggregation through concatenation or addition, (b) top-down feature aggregation with FPM

fuses features extracted from different stages in the backbone network, where the top-down pathway upsamples the feature map from the higher pyramid level to produce high-resolution features and then enhances them with features from the backbone network by elementwise addition. Each fused feature is filtered by a convolutional layer and pooled by a temporal pooling layer to generate embeddings at different pyramid stages. Then, these embeddings are concatenated to obtain more discriminative speaker embedding.

Table 1: The architecture of the backbone ResNet34. Inside the brackets is the shape of a residual block, and outside the brackets is the number of stacked blocks on a stage. The input size is $T \times 64$.

Stage	Layer name	Output size	ResNet34
-	Conv0	$T \times 64 \times 32$	$7 \times 7, 32, \text{stride } 2$
1	Conv1_x	$T/2 \times 32 \times 32$	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$
2	Conv2_x	$T/2 \times 16 \times 64$	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 4$
3	Conv3_x	$T/2 \times 8 \times 128$	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 6$
4	Conv4_x	$T/2 \times 4 \times 256$	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$

3. Proposed Method

In this section, we discuss details of the proposed bidirectional multiscale feature aggregation (BMFA) network along with the attentional fusion module.

3.1. Bidirectional Multiscale Feature Aggregation

Fig. 2 (a) schematically depicts the overall architecture of the proposed network. The network consists of three branches: a backbone branch, a bottom-up aggregation branch, and a top-down aggregation branch.

Backbone branch: We utilize ResNet34 as the backbone model in this work. The detailed configurations of ResNet34

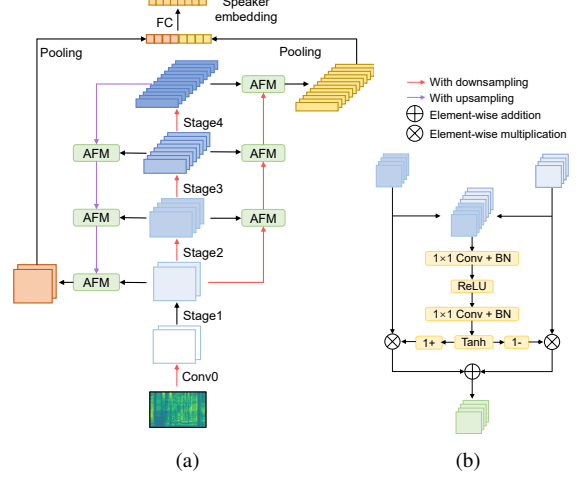


Figure 2: (a) Illustration of the proposed bidirectional multiscale aggregation architectures. (b) The structure of the attentional fusion module

are listed in Table 1. We extract feature maps with different resolutions from the last layer of every stage and denote the feature map of the i -th stage as C_i for $i=1,2,3,4$. Then these features are transmitted to both top-down and bottom-up aggregation branches separately.

Top-down aggregation branch: By propagating high-level semantic context from top layers into bottom layers, top-down aggregation can help to suppress noisy information in bottom layers. Specifically, each feature map (F_i^{tb} , $i = 3, 2, 1$) of the top-down branch is iteratively built by combining the backbone feature map (C_i) from the same stage and the higher-stage top-down feature map (F_{i+1}^{tb}):

$$\begin{cases} F_4^{tb} = C_4 \\ F_i^{tb} = AFM(\mathcal{U}(\mathcal{B}(W_i^{tb} * F_{i+1}^{tb}), \mathcal{B}(W_i^{l_{ctb}} * C_i))) \end{cases} \quad (1)$$

where $\mathcal{B}(\cdot)$ denotes batch normalization [21] and $\mathcal{U}(\cdot)$ is the upsampling operation in the frequency dimension with a factor of 2. $*$ denotes the convolution operation, and $AFM(\cdot)$ is operator of the proposed attentional fusion module (see section 3.2). W_i^{tb} and W_i^{lc} are both 1×1 convolution filters, where W_i^{tb} is used to reduce the channel number to be same with C_i , while $W_i^{l_{ctb}}$ is used in lateral connection to re-encode the channel information of feature map C_i . Additionally, we append a 3×3 convolutional layer on the bottom feature map F_1^{tb} to refine the features, followed by statistical pooling layer to encapsulate the variable size features into a fixed size embedding h^{tb} .

Bottom-up aggregation branch: In the bottom-up branch, the complementary low-level details from the bottom layers are fused into high-level features. Specifically, each feature map (F_i^{bt} , $i = 2, 3, 4$) of the bottom-up branch is iteratively built by combining the backbone feature map (C_i) from the same stage and the lower-stage bottom-up feature map (F_{i-1}^{bt}):

$$\begin{cases} F_1^{bt} = C_1 \\ F_i^{bt} = AFM(\mathcal{B}(W_i^{bt} * \mathcal{D}(F_{i-1}^{bt})), \mathcal{B}(W_i^{l_{c_{bt}} * C_i})) \end{cases} \quad (2)$$

where $\mathcal{D}(\cdot)$ denotes the downsampling operation in the frequency dimension with a factor of 2. W_i^{bt} and $W_i^{l_{c_{bt}}}$ are both

1×1 convolution filters, where \mathbf{W}_i^{tb} is used to increase the channel number to be same with \mathbf{C}_i , and $\mathbf{W}_i^{lc_{bt}}$ has the same purpose as $\mathbf{W}_i^{lc_{tb}}$. Similar to the top-down branch, we append a 3×3 convolutional layer on the top feature map \mathbf{F}_1^{bt} , followed by a statistic pooling layer to generate utterance-level bottom-up embedding \mathbf{h}^{bt} .

Finally, we perform a fully connected layer on the concatenation of \mathbf{h}^{bt} and \mathbf{h}^{tb} to achieve a nonlinear transformation and dimension reduction to obtain speaker embedding for verification. In the training phase, an additional fully connected layer and softmax layer are appended to the speaker embeddings for the speaker classification task.

3.2. Attentional Fusion Module

Existing MFA architectures rarely focus on feature fusion modules and usually employ simple operations such as elementwise addition or concatenation, which are suboptimal for combining feature maps of different importance. To address this problem, we design an attentional fusion module (AFM) to compute attention weights.

As shown in Figure 2 (b), AFM takes the concatenation of two feature maps as the input and computes the time-frequency attention map using two convolutional layers. Given two feature maps \mathbf{X}, \mathbf{Y} with C channels, the attention map \mathbf{S} is computed by:

$$\mathbf{S} = \tanh(\mathbf{B}((\mathbf{W}_2 * \text{ReLU}(\mathbf{B}(\mathbf{W}_1 * [\mathbf{X}, \mathbf{Y}])))) \quad (3)$$

where $[\cdot]$ refers to the concatenation along the channel dimension. \mathbf{W}_1 and \mathbf{W}_2 are 1×1 convolution filters for aggregating local channel context information with output channel sizes of C/r and C , respectively. r is the channel reduction ratio (we set $r=4$ in this work). Note that \mathbf{S} has the same shape as \mathbf{X} and \mathbf{Y} . Then, we fuse feature maps \mathbf{X} and \mathbf{Y} using the attention map as:

$$\mathbf{F} = \text{AFM}(\mathbf{X}, \mathbf{Y}) = (1 \oplus \mathbf{S}) \otimes \mathbf{X} + (1 \ominus \mathbf{S}) \otimes \mathbf{Y} \quad (4)$$

where \oplus and \ominus denote broadcasting addition and subtraction, respectively. \otimes denotes elementwise multiplication. In AFM, \mathbf{S} represents the correlation between inputs \mathbf{X} and \mathbf{Y} . After broadcasting addition and subtraction on \mathbf{S} , the fusion weights of \mathbf{X} and \mathbf{Y} are complementary to each other, which enables module to conduct a soft aggregation of \mathbf{X} and \mathbf{Y} by enhancing one and suppressing the other.

4. Experimental Setup

4.1. Datasets and evaluation metrics

The experiments are carried out on the NIST SRE16 evaluation dataset, and the VoxCeleb1 dataset. For the NIST SRE16 dataset, the training data mainly consist of telephone speech (with a small amount of microphone speech) from the NIST SRE2004-2010, Mixer 6 and Switchboard datasets. We also use the data augmentation techniques described in [4], which employ the babble, music, noise and reverb augmented data to increase the quantity and diversity of the existing training data. In summary, there were a total of 183,457 recordings from 7,001 speakers, including approximately 96,000 randomly selected augmented recordings. The performance is evaluated in terms of equal error rate (EER), the minimal detection cost function (minDCF) and actual detection cost function (actDCF) calculated using the SRE16 official scoring software. For the NIST SRE16, the equalized results are used.

For the VoxCeleb1 test set, we use the development part of the VoxCeleb2 dataset [12] without any data augmentation as training data, which contains 5994 speakers with 1,092,009 utterances. The performance is evaluated in terms of EER and minDCF with $P_{\text{target}} = 10^{-2}$.

4.2. Input features

The input acoustic features are 64-dimensional FBanks with a frame length of 25 ms that are mean-normalized over a sliding window of up to 3 s. An energy-based voice activity detector (VAD) is used to remove all the silent frames. The utterances are cut into chunks with durations ranging from 200 to 400 frames for training.

4.3. Model configuration

In this work, we implement all models in TensorFlow toolkit [22]. The other procedures (including data processing, feature extraction and the PLDA backend) are implemented using the Kaldi toolkit [23]. The network is optimized using the Adam optimizer with minibatch size of 64, and the learning rate gradually decreases from $1e-3$ to $1e-4$. We use bilinear interpolation to upsample the feature maps in the top-down branch and 3×3 convolution with a stride of 2 in the frequency dimension to downsample the feature maps in the bottom-up branch. Both fully connected layers after the statistic pooling layer have 512 nodes. The additive margin softmax (AM-softmax) [24] is used as the training criterion for all models, and the margin m and scaling factor s are set to 0.15 and 30, respectively. We build four systems for comparison, and the configurations of each system are listed as follows:

1. Baseline: ResNet34 network.
2. MFA.S34: ResNet34 network with MFA as described in [14] (Fig.1 (a)), aggregating features of stage 3 and stage 4.
3. MEA.FPM: ResNet34 network with multiscale embedding aggregation (MEA) using FPM as described in [17] (Fig.1 (b)).
4. BMFA-AFM: the proposed network in this paper, i.e., the ResNet34 network with bidirectional multiscale feature aggregation (BMFA) using the attentional fusion module (AFM) as described in section 3 (Fig. 2 (a)).

4.4. PLDA Backend

For the NIST SRE16, the DNN embeddings are centered using the unlabeled development data and are projected using LDA, which reduces the dimensions of the speaker embedding to 100. The PLDA model is trained and then adapted to the unlabeled data through unsupervised adaptation in Kaldi. For VoxCeleb1, no adaptation technique or LDA projection is used.

5. Results and Analysis

5.1. Results on the SRE16 dataset

5.1.1. Comparison between different aggregation strategies

Table 2 presents the performance of the systems with different feature aggregation strategies on the NIST SRE16 dataset. Furthermore, we compare the proposed attentional fusion module (AFM) with the conventional elementwise addition and concatenation fusion methods in Table 2.

Table 2: Comparison results of different multiscale feature aggregation strategies and different fusion methods on the NIST SRE16. For the fusion module, Add and Concat denote elementwise addition and concatenation operation, respectively.

Aggregation strategy	Fusion method	EER(%)	Pooled		EER(%)	Cantonese		EER(%)	Tagalog	
			minDCF	actDCF		minDCF	actDCF		minDCF	actDCF
Baseline	-	6.54	0.5505	0.5671	2.99	0.3017	0.3225	10.05	0.7357	0.8118
MFA_S34	Concat [14]	6.34	0.5348	0.5482	2.87	0.2897	0.3036	9.84	0.7218	0.7928
MFA_S34	AFM	6.21	0.5118	0.5191	2.70	0.3033	0.3215	9.66	0.6889	0.7166
MEA_FPM	Add [17]	6.33	0.5031	0.5082	2.53	0.2832	0.3000	10.06	0.6963	0.7164
MEA_FPM	AFM	6.10	0.4926	0.4981	2.50	0.2786	0.3022	9.61	0.6757	0.6939
BMFA	Concat	6.23	0.5001	0.5037	2.66	0.2694	0.2917	9.82	0.6952	0.7158
BMFA	Add	6.00	0.4964	0.5017	2.51	0.2797	0.2938	9.63	0.6856	0.7096
BMFA	AFM	5.79	0.4620	0.4673	2.39	0.2531	0.2843	9.20	0.6438	0.6503

It can be observed that the systems with the multiscale aggregation strategy outperform the ResNet34 baseline system in all evaluation conditions. Additionally, we observe that the proposed attentional fusion module (AFM) can obtain consistent performance improvement over the conventional fusion methods for almost all three aggregation strategies. Among all of the abovementioned aggregation strategies, our BMFA-AFM system, employing bidirectional feature aggregation and attentional fusion modules, achieves the best performance, which improves the EER, minDCF and actDCF over the baseline system on the pooled set by 11.5%, 16.1% and 17.6%, respectively.

5.1.2. Ablation study

To investigate the effect of (1) the bidirectional aggregation strategy and (2) the features from different stages of the backbone network for the proposed BMFA-AFM model, we perform an ablation study, as shown in Table 3. For simplicity, only the SRE16 pooled results are listed. In Table 3, MFA_Top-down, MFA_Bottom-up denote multiscale feature aggregation using features from stage 1 to stage 4 with only top-down or bottom-up branch (described as section 3.1), respectively. BMFA_S34 refers to BMFA using features from stage 3 and stage 4, and BMFA_S234 refers to BMFA using features from stage 2, 3 and 4. BMFA_S1234 refers to BMFA using features of stage 1 to 4, namely BMFA-AFM. All models apply AFM for feature fusion.

We can observe from the first set of experiments ('+MFA_Top-down' and '+MFA_Bottom-up') that both top-down and bottom-up aggregation strategies can boost the performance over the baseline. Compared to the top-down aggregation strategy, the bottom-up strategy obtains a more obvious improvement. We can observe that combining the bottom-up and top-down aggregation (BMFA) further improves the performance by a large margin. Furthermore, the performance can be improved gradually by adding more features from lower stages for BMFA systems, as shown in the bottom three rows in Table 3. This consistent improvement indicates that features of lower stages can provide extra complementary information for the final verification task.

5.2. Comparison with recent methods on Voxceleb1

In this section, we compare the proposed method with several recent approaches on the VoxCeleb1 dataset. As shown in Table 4, with the same ResNet-based feature extractor, the proposed method achieves the best results among all listed methods. Compared to the recent multiscale aggregation methods (ResNetXt41-MSA and ResNet34-FPM), our model outperforms them by a large margin. Specifically, the proposed method achieves the best results, with EER of 2.98% and minDCF of 0.3008 when training on VoxCeleb1 dev and EER

of 1.73% and minDCF of 0.1873 when training on VoxCeleb2 dev. The results demonstrate that the proposed bidirectional aggregation strategy with attentional fusion module is a more efficient method for the aggregation of multiscale features.

Table 3: Ablation study for different directional aggregation and different features.

Aggregation strategy	EER(%)	minDCF	actDCF
Baseline	6.54	0.5505	0.5671
+MFA_Top-down	6.33	0.5215	0.5296
+MFA_Bottom-up	6.12	0.5092	0.5150
+BMFA_S34	6.34	0.4980	0.5036
+BMFA_S234	6.09	0.4822	0.4911
+BMFA_S1234	5.79	0.4620	0.4673

Table 4: Results on the VoxCeleb1 dataset. LDE denotes learnable dictionary encoding [13], and SPE denotes spatial pyramid encoding [25]

System	Train set	EER(%)	minDCF
ResNet34-LDE [13]	Vox1	4.56	0.441
ResNetXt41-MFA [14]	Vox1	4.09	0.4578
ResNet34-SPE [25]	Vox1	4.03	0.4020
ResNet34-FPM [17]	Vox1	3.22	0.3500
ResNet34-BMFA-AFM(ours)	Vox1	2.98	0.3008
ResNet50 [12]	Vox2	3.95	0.429
ThinResNet34-GhostVLAD [26]	Vox2	3.22	N/A
ResNet34-SPE [25]	Vox2	2.61	0.245
ResNet34-FPM [17]	Vox2	1.98	0.2050
ResNet34-BMFA-AFM (ours)	Vox2	1.73	0.1873

6. Conclusions

In this paper, we proposed a novel MFA method with bidirectional aggregation strategy and attentional fusion module for speaker verification. The bidirectional aggregation strategy combines and refines the feature maps from different stages iteratively in both a bottom-up and top-down manner for learning better speaker embedding. Applying the attentional fusion module can significantly improve the performance of BMFA by computing the optimal fusion weights for input feature maps, which is also effective for other MFA strategies. Experiments conducted on the SRE16 and VoxCeleb1 datasets demonstrate that our proposed method outperforms existing MFA methods for the SV task.

7. Acknowledgements

This work was partially funded by the National Natural Science Foundation of China (Grant No. U1836219).

8. References

- [1] 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*, vol. 19, no. 4, pp. 788–798, 2010.
- [2] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Odyssey*, vol. 14, 2010.
- [3] 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 (ICASSP)*. IEEE, 2014, pp. 4052–4056.
- [4] 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 (ICASSP)*. IEEE, 2018, pp. 5329–5333.
- [5] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Interspeech*, 2017, pp. 1487–1491.
- [6] Y. Jiang, Y. Song, I. McLoughlin, Z. Gao, and L.-R. Dai, "An effective deep embedding learning architecture for speaker verification," in *INTERSPEECH*, 2019, pp. 4040–4044.
- [7] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapattnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.
- [8] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.
- [9] Z. Gao, Y. Song, I. V. McLoughlin, W. Guo, and L.-R. Dai, "An improved deep embedding learning method for short duration speaker verification," 2018.
- [10] W. Lin and M.-W. Mak, "Wav2spk: A simple dnn architecture for learning speaker embeddings from waveforms," *Proc. Interspeech 2020*, pp. 3211–3215, 2020.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [12] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.
- [13] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," *arXiv preprint arXiv:1804.05160*, 2018.
- [14] Z. Gao, Y. Song, I. McLoughlin, P. Li, Y. Jiang, and L.-R. Dai, "Improving aggregation and loss function for better embedding learning in end-to-end speaker verification system," in *INTERSPEECH*, 2019, pp. 361–365.
- [15] S. Seo, D. J. Rim, M. Lim, D. Lee, H. Park, J. Oh, C. Kim, and J.-H. Kim, "Shortcut connections based deep speaker embeddings for end-to-end speaker verification system," in *INTERSPEECH*, 2019, pp. 2928–2932.
- [16] A. Hajavi and A. Etemad, "A deep neural network for short-segment speaker recognition," *arXiv preprint arXiv:1907.10420*, 2019.
- [17] Y. Jung, S. M. Kye, Y. Choi, M. Jung, and H. Kim, "Improving multi-scale aggregation using feature pyramid module for robust speaker verification of variable-duration utterances," *arXiv preprint arXiv:2004.03194*, 2020.
- [18] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, "Attentional feature fusion," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3560–3569.
- [19] S. O. Sadjadi, T. Kheyrkhan, A. Tong, C. S. Greenberg, D. A. Reynolds, E. Singer, L. P. Mason, and J. Hernandez-Cordero, "The 2016 nist speaker recognition evaluation," in *Interspeech*, 2017, pp. 1353–1357.
- [20] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.
- [21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.
- [22] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.
- [23] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.
- [24] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.
- [25] Y. Jung, Y. Kim, H. Lim, Y. Choi, and H. Kim, "Spatial pyramid encoding with convex length normalization for text-independent speaker verification," *arXiv preprint arXiv:1906.08333*, 2019.
- [26] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5791–5795.