INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Branch-ECAPA-TDNN: A Parallel Branch Architecture to Capture Local and Global Features for Speaker Verification

Jiadi Yao[1,†], *Chengdong Liang*[1,2,†], *Zhendong Peng*[2], *Binbin Zhang*[2], *Xiao-Lei Zhang*[1,3,⋆]

[1]School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China
[2]Horizon Robotics, Beijing, China
[3]Research and Development Institute of Northwestern Polytechnical University in Shenzhen, China

yaojiadi@mail.nwpu.edu.cn, xiaolei.zhang@nwpu.edu.cn

Abstract

Currently, ECAPA-TDNN is one of the state-of-the-art deep models for automatic speaker verification (ASV). However, it focuses too much on local feature extraction with fixed local ranges, without paying much attention to global feature extraction. To deal with this issue, in this paper, we propose Branch-ECAPA-TDNN, which uses two parallel branches to extract features with various ranges and abstract levels. One branch employs multi-head self-attention to capture long-range dependencies, while the other branch utilizes an SE-Res2Block module to model local multi-scale characteristics. To improve the feature fusion, we further apply different merging methods to aggregate features from both branches. Experimental results demonstrate that the proposed Branch-ECAPA-TDNN achieves a relative EER reduction of 24.10% and 7.92% over ECAPA-TDNN on the VoxCeleb and CN-Celeb datasets, respectively.

Index Terms: speaker verification, self-attention, x-vector, Res2Net, parallel branch

1. Introduction

Automatic speaker verification (ASV) is a task of verifying whether an utterance is pronounced by a claimed speaker. In recent years, ASV has been rapidly developed [1, 2, 3], and finds its wide applications in intelligent housing systems, voice-based authentication, bank trading and remote payment. In general, the state-of-the-art research on ASV contains two components. The first one is the embedding extractor [4, 5, 6], which aims to extract speaker embeddings with a fixed-dimension from utterances to represent the acoustic characteristics of speakers, where deep-learning-based embedding extractors [7] reach the state-of-the-art performance. The other one is the scoring back-end, which aims to calculate the similarity between two speaker embedding vectors. The most common back-ends are the cosine similarity scoring and probabilistic linear discriminant analysis [8].

Convolution neural networks are the most favorite for ASV. For example, the x-vector [4, 5] based on one-dimensional (1D) convolution is the most prevalent embedding extractor for ASV. Recently, various network structures have been developed for the speaker embedding extraction, including the time-delay neural network (TDNN) [5, 9], ResNet [10, 11, 12, 13], and their variations [2, 14, 15, 16]. It is worthy noting that the ECAPA-TDNN [6] and its extensions [17, 18], which integrate the building blocks of TDNN and squeeze-and-excitation (SE) modules [19] with Res2Block [20], achieve the state-of-the-art performance.

† Equal Contribution.
⋆ Corresponding author.

However, ECAPA-TDNN still has limitations. It mainly focuses on local feature modeling, which lacks the ability of global feature fusion. Its convolution kernel has a fixed size, which makes its receptive field unable to capturing the global temporal and frequency speaker patterns efficiently. The above weaknesses make the extracted speaker representation do not contain important global context information. To address this issue, many Transformer-based models [21, 22, 23] have been introduced, where the multi-head self-attention mechanism is good at capturing long-range dependencies. However, we believe that the performance of the Transformer-based ASV still has much room of improvement.

In this paper, to further address the weaknesses of ECAPA-TDNN, we propose the *Branch-ECAPA-TDNN* framework. It has two parallel branches for capturing speaker information in both the global range and various local ranges, one of which employs multi-head self-attention to capture long-range dependencies, and the other utilizes an SE-based Res2Block (SE-Res2Block) module to extract local relationships. In addition, inspired by [24], we employ multiple merging mechanisms to merge the output of the two branches for further improving the performance of Branch-ECAPA-TDNN. Our contribution includes:

- We proposed a new ASV model, named *Branch-ECAPA-TDNN*. It extends the current state-of-the-art ECAPA-TDNN with two branches built on convolution and self-attention operators respectively for learning both local and global information.
- We merge the branches of Branch-ECAPA-TDNN by multiple merging mechanisms to mine the local and global information in depth.
- We conducted extensive experiments on the VoxCeleb and CN-Celeb datasets, respectively. Extensive experiments demonstrate that the proposed method achieves a relative EER reduction of 24.10% and 7.92% over ECAPA-TDNN on the VoxCeleb and CN-Celeb datasets, respectively.

2. Proposed Methods

In this section, we present the framework and fundamental components of the proposed Branch-ECAPA-TDNN.

2.1. Framework

The framework of Branch-ECAPA-TDNN is shown in Figure 1, where BN denotes Batch Normalization, FC denotes Fully Connected Layer, and the non-linearities are Rectified Linear Units (ReLU) unless otherwise stated. The proposed Branch-ECAPA-TDNN framework is analogous to ECAPA-TDNN [6]. The fundamental difference lies that it employs a Branch block

10.21437/Interspeech.2023-402

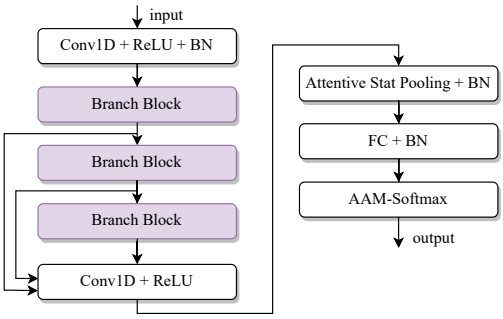

Figure 1: *Architecture of Branch-ECAPA-TDNN.*

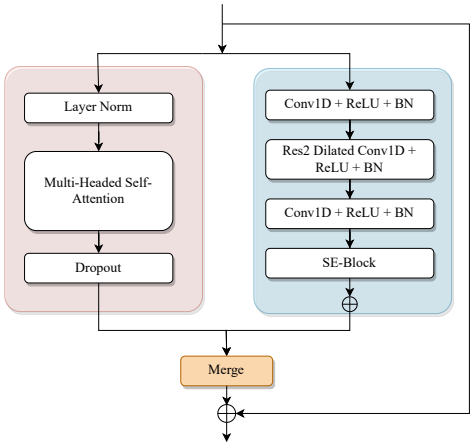

Figure 2: *Architecture of the Branch block in Branch-ECAPA-TDNN. It consists of two parallel branches. One branch uses attention to capture global information, while the other branch uses SE-Res2Block to extract local information.*

instead of the SE-Res2Block module in ECAPA-TDNN to capture both local and global speaker characteristics. The detailed structure of the proposed Branch block is illustrated in Figure 2. It is composed of two parallel branches and a merging module, where one of the branch is a self-attention branch for capturing global features, and the other one is a SE-Res2Block branch for extracting local features. We will introduce the three components in the following subsections.

2.2. The attention branch for global feature modeling

In Figure 2, the left branch is the multi-head self-attention module [25], which aims to extract the global speaker feature. We describe the branch in detail as follows.

Let $\mathbf{X} \in \mathbb{R}^{T\times D}$ denotes the input, where T and D represent the number of time frames and the dimension of the acoustic features, respectively. Assuming the number of the attention heads of the self-attention is h, then, for each head, the input feature $\mathbf{X}$ is projected into the query (Q), key (K) and value (V) subspaces of dimension E as follows:

$$\mathbf{Q}^i = \mathbf{X}\mathbf{W}_Q^i, \mathbf{K}^i = \mathbf{X}\mathbf{W}_K^i, \mathbf{V}^i = \mathbf{X}\mathbf{W}_V^i. \tag{1}$$

where $\mathbf{Q}^i$, $\mathbf{K}^i$ and $\mathbf{V}^i$ denotes the query, key and value embeddings of the i-th attention head, respectively, all of which are in $\mathbb{R}^{T\times d_k}$, $\mathbf{W}_j^i \in \mathbb{R}^{D\times d_k}$ ($\forall j \in Q, K, V$, $d_k = E/h$) are the linear projection parameters. We compute the dot products of the query with all keys, divide the result of each dot product by $\sqrt{d_k}$ which is further applied with the softmax function for obtaining an attention matrix $\mathbf{Z}^i \in \mathbb{R}^{T\times d_k}$:

$$\mathbf{Z}^i = \text{softmax}\left(\frac{\mathbf{Q}^i \cdot \left(\mathbf{K}^i\right)^\top}{\sqrt{d_k}}\right)\mathbf{V}^i. \tag{2}$$

Finally, the outputs of all attention heads are concatenated across the subspaces and transformed to the original size by:

$$\mathbf{Y}_\text{A} = \text{concat}\left[\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^h\right]\mathbf{W}_O. \tag{3}$$

where $\mathbf{Y}_\text{A} \in \mathbb{R}^{T\times D}$, $\mathbf{W}_O \in \mathbb{R}^{E\times D}$ is a parameter matrice of the projection layer.

2.3. SE-Res2Block for local feature modeling

In Figure 2, the right branch is a SE-Res2Block module [6], which integrates the Res2Net [20, 15] module and the Squeeze-and-Excitation [19] block to further represent multi-scale features with various granularity.

First, for the Res2Net module, we divide the input feature maps generated by 1×1 convolution into s subsets $\{\mathbf{x}_i\}_{i=1}^s$, where all channels have the same spatial size, and each channel occupies $1/s$ of the channels of the input feature maps, i.e. $\mathbf{x}_i \in \mathbb{R}^{T\times D/s}$. The 3×3 convolution, denoted by $\mathbf{K}_i$, is applied to each subset, except $\mathbf{x}_1$, in a hierarchical residual-style connection. Specifically, after applying the convolution to $\mathbf{x}_{i-1}$, the output of $\mathbf{K}_{i-1}$ is added with $\mathbf{x}_i$ before going through $\mathbf{K}_i$. The above process can be described formally by:

$$\mathbf{m}_i = \begin{cases} \mathbf{x}_i, & i = 1; \\ \mathbf{K}_i\left(\mathbf{x}_i\right), & i = 2; \\ \mathbf{K}_i\left(\mathbf{x}_i + \mathbf{m}_{i-1}\right), & i = 3, 4, \ldots, s. \end{cases} \tag{4}$$

This process further expands the potential receptive fields of a layer, leading to multiple diverse feature scales. The output of this module $\{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_s\}$ are concatenated and then fed into a subsequent 1×1 convolutional layer to generate $\mathbf{M}$ for fusing the information from different scales.

Next, the output $\mathbf{M} \in \mathbb{R}^{T\times D}$ goes through a squeeze-and-excitation block. Specifically, the squeeze operation obtains a squeeze vector $\mathbf{u}$ by performing a global average pooling:

$$\mathbf{u} = \frac{1}{T}\sum_t^T M_t. \tag{5}$$

where M_t is the t-th frame of $\mathbf{M}$. The excitation operation gets the weight of each channel by:

$$\mathbf{q} = \sigma\left(\mathbf{W}_2\left(\text{ReLU}\left(\mathbf{W}_1\mathbf{u}\right)\right)\right). \tag{6}$$

where $\sigma(\cdot)$ denotes the sigmoid function, $\mathbf{W}_1 \in \mathbb{R}^{B\times D}$, $\mathbf{W}_2 \in \mathbb{R}^{D\times B}$, with B denoted as the number of dimensions within the bottleneck layer.

Finally, each dimension of $\mathbf{M}$, denoted as $\mathbf{M}_i$, is rescaled by:

$$\mathbf{Y}_{\text{R}i} = q_i\mathbf{M}_i, \quad \forall i = 1, 2, \ldots, D. \tag{7}$$

where q_i is the i-th element of $\mathbf{q}$. We further concatenate all $\mathbf{Y}_{\text{R}i}$ into a matrix $\mathbf{Y}_\text{R} = [\mathbf{Y}_{\text{R}1}, \mathbf{Y}_{\text{R}2}, \ldots, \mathbf{Y}_{\text{R}D}]$.

2.4. Merging methods

In this section, inspired by [24], we employ three merging mechanisms to fuse both the local and global features.

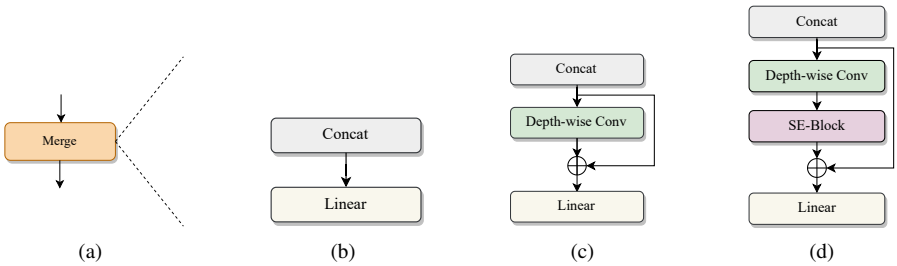

Figure 3: *Architecture of the proposed three merging methods. (a) The framework of the merging module. (b) The concatenation-based merging method. (c) The depth-wise-convolution-based merging method. (d) The squeeze-and-excitation-based merging method.*

2.4.1. Concatenation

As shown in Figure 3(b), we concatenate $\mathbf{Y}_{\mathrm{A}}$ and $\mathbf{Y}_{\mathrm{R}}$ and then project the concatenated feature back to their original dimension:

$$\mathbf{Y}_{\mathrm{Merge}} = \mathrm{Concat}(\mathbf{Y}_{\mathrm{A}}, \mathbf{Y}_{\mathrm{R}})\mathbf{W}_{\mathrm{con}}, \tag{8}$$

where $\mathbf{W}_{\mathrm{con}} \in \mathbb{R}^{2D\times D}$ is a learnable parameter matrix of the linear projection.

2.4.2. Depth-wise convolution

As shown in Figure 3(c), we incorporate the depth-wise convolution [26, 27] into the merging module, which makes it able to use information from adjacent features when integrating features from two branches. Specifically, we first concatenate the outputs of the two branches, $\mathbf{Y}_{\mathrm{A}}$ and $\mathbf{Y}_{\mathrm{R}}$, to obtain $\mathbf{Y}_{\mathrm{C}}$. Then we use a depth-wise convolution to generate $\mathbf{Y}_{\mathrm{D}}$, aiming to enhance the spatial information exchange process. Finally, we use a residual connection. The detailed process is as follows:

$$\begin{aligned} \mathbf{Y}_{\mathrm{C}} &= \mathrm{Concat}(\mathbf{Y}_{\mathrm{A}}, \mathbf{Y}_{\mathrm{R}}), \\ \mathbf{Y}_{\mathrm{D}} &= \mathrm{DwConv}(\mathbf{Y}_{\mathrm{C}}), \\ \mathbf{Y}_{\mathrm{Merge}} &= (\mathbf{Y}_{\mathrm{C}} + \mathbf{Y}_{\mathrm{D}})\mathbf{W}_{\mathrm{dw}}. \end{aligned} \tag{9}$$

where DwConv denotes the depth-wise convolution, $\mathbf{W}_{\mathrm{dw}} \in \mathbb{R}^{2D\times D}$ is a learnable parameter matrix of the linear projection.

2.4.3. Squeeze-and-excitation operation

As shown in Figure 3(d), we add a SE module right after the depth-wise convolution, which is similar to the squeeze-and-excitation operation in SE-Res2Block mentioned in Section 2.3, except that we replace the ReLU activation function in (6) with the Swish [28] non-linear function. In the merging process, introducing the SE operation can extract the global information extensively, and enhance the output of the depth-wise convolution.

3. Experiments

In this section, we present datasets, implementation details, evaluation protocols, and experimental results.

3.1. Dataset

The experiments were conducted on the VoxCeleb [29, 13] and CN-Celeb [30, 31] datasets.

For the experiments on the VoxCeleb, we trained the speaker verification models on the development set of VoxCeleb2 [13], which contains 1,092,009 utterances among 5,994 speakers. The development set and test set of VoxCeleb1 [29] were used for the evaluation. There are three types of evaluation trials, which are VoxCeleb1-O, VoxCeleb1-E and VoxCeleb1-H.

For the experiments on CN-Celeb, we used 1,966 speakers from CN-Celeb2 and 797 speakers from the development set of CN-Celeb1 together as the training set, and conducted the evaluation on the test set of CN-Celeb1.

Online data augmentation [32] was used during training. The noise datasets in the data augmentation are from MUSAN [33] and RIRs [34]. In addition, we applied speed perturbation to data augmentation, where we randomly altered the speed of an utterance at a ratio selected randomly from $\{0.9, 1.0, 1.1\}$.

3.2. Implementation details

We used *WeSpeaker*[1] toolkit to implement the proposed Branch-ECAPA-TDNN and replicated the baseline ECAPA-TDNN. The input features were 80-dimensional log mel-filter banks (Fbank) pre-emphasized by a Hamming window with a window length of 25ms and a window shift of 10ms. All training data were chunked into 200 frames. Each chunk was normalized by the Cepstral mean normalization. All models were trained using the AAM-Softmax [36] loss, where the scale was 32, the initial margin was 0 and the final margin was 0.2. We used the margin schedule in [37] to update the margin.

The learning rate schedule in [37], which contains warm-up and exponential decrease strategies, was used to update the learning rate. The initial learning rate was 0.1, and the final learning rate was 5e-5. Following [6], we set the bottleneck dimension B in the SE-Block to 128, the scale dimension s in the Res2Net module to 8, and the number of nodes of the final fully connected layer to 192.

Besides the ECAPA-TDNN baseline, we also used TDNN [5], Extended-TDNN (E-TDNN) [9, 35] and ResNet [11] as our baselines, whose experimental results were copied from [6].

3.3. Evaluation protocol

In the test phase, we employed cosine similarity as the scoring criterion. The adaptive score normalization (AS-Norm) [38, 37] was applied to normalize the scores. We used the top n_{top} cohort segments for the score normalization. We adopted the standard equal error rate (EER) and the minimum detection cost function (minDCF) with $P_{\mathrm{target}} = 0.01$ and $C_{\mathrm{miss}} = C_{\mathrm{fa}} = 1$, as the evaluation protocols.

[1] https://github.com/wenet-e2e/wespeaker

Table 1: *EER (%) and minDCF of the comparison methods on the VoxCeleb and CN-Celeb datasets, where the parameter C denotes the number of filters in the convolutional layer of SE-Res2Block. The marker "(b)" "(c)" and "(d)" attached with Branch-ECAPA-TDNN refers to the merging methods in Figure 3. The term "AS-Norm300" denotes that n_{top}=300 in AS-Norm.*

Architecture	# Params	VoxCeleb1-O		VoxCeleb-E		VoxCeleb-H		CN-Celeb	
		EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF
TDNN [5]	4.61M	2.016	0.191	1.949	0.228	3.476	0.332	9.772	0.556
E-TDNN [9]	6.80M	1.490	0.160	1.610	0.171	2.690	0.242	-	-
E-TDNN (large) [35]	20.40M	1.260	0.140	1.370	0.149	2.350	0.215	-	-
ResNet18 [11]	13.80M	1.470	0.177	1.600	0.179	2.880	0.267	-	-
ResNet34 [11]	23.90M	1.190	0.159	1.330	0.156	2.460	0.229	-	-
C=512									
ECAPA-TDNN	6.19M	1.191	0.114	1.254	0.139	2.285	0.219	8.313	0.432
+ AS-Norm300		0.979	0.124	1.157	0.126	2.065	0.198	7.644	0.390
Branch-ECAPA-TDNN(b)	9.34M	0.904	**0.094**	1.129	0.126	2.126	0.214	7.716	0.412
+ AS-Norm300		0.862	0.103	1.058	0.125	1.957	0.193	**7.215**	**0.371**
Branch-ECAPA-TDNN(c)	9.36M	0.941	0.111	1.102	0.125	2.075	0.204	7.655	0.416
+ AS-Norm300		**0.808**	0.103	1.045	**0.118**	1.923	**0.186**	7.232	**0.371**
Branch-ECAPA-TDNN(d)	10.14M	1.016	0.104	1.095	0.126	2.065	0.205	7.910	0.415
+ AS-Norm300		0.872	0.103	**1.031**	**0.118**	**1.919**	**0.186**	7.350	0.373
C=1024									
ECAPA-TDNN	14.65M	0.920	0.103	1.064	0.117	2.006	0.194	7.879	0.420
+ AS-Norm300		0.782	0.119	0.979	0.108	1.801	0.180	7.412	0.379
Branch-ECAPA-TDNN(b)	24.11M	0.808	0.091	0.982	0.107	1.853	0.182	7.339	0.397
+ AS-Norm300		**0.718**	**0.084**	0.916	**0.098**	**1.690**	**0.166**	6.978	0.358
Branch-ECAPA-TDNN(c)	24.13M	0.814	0.098	0.970	0.112	1.903	0.184	7.519	0.396
+ AS-Norm300		0.728	0.100	0.922	0.103	1.776	0.177	6.984	**0.352**
Branch-ECAPA-TDNN(d)	25.71M	0.808	0.090	0.968	0.111	1.888	0.193	7.401	0.398
+ AS-Norm300		0.755	0.104	**0.899**	0.107	1.741	0.179	**6.922**	0.357

3.4. Main results

Table 1 lists the performance of the proposed Branch-ECAPA-TDNN and the baseline systems. From the table, we see that the proposed method outperforms the baseline systems, which indicates that the global modeling capability of Branch-ECAPA-TDNN is significantly improved over its counterpart ECAPA-TDNN by applying the multi-head self-attention mechanism to extract global speaker characteristics. Moreover, although the two branches share the same input, they focus on different scope of the spatial relationships, thus achieving complementary advantages of each other. Specially, compared with ECAPA-TDNN, the proposed Branch-ECAPA-TDNN achieves up to a relative EER reduction of 24.10% on the VoxCeleb dataset and up to a relative EER reduction of 7.92% on the CN-Celeb dataset, without using the AS-Norm. Third, when the parameter C increases, the complexity of the network increases, which improves the performance, with a negative effect of bringing larger parameter calculation and information redundancy. Finally, the results with AS-Norm demonstrate that the score normalization by reducing within trial variability leads to higher performance and better calibration.

3.5. Effects of different merging mechanisms

This section discusses the effects of different merging mechanisms in Figure 3. Table 1 lists the comparison results of the merging methods. From the table, we observe the following phenomena. First, the performance improvement of the concatenation-based method whose architecture is drawn in Figure 3(b), is limited at a small parameter scale $C = 512$. This is mainly caused by that the concatenation-based merging simply concatenates the output information of the local and global extractors without exchanging the information from adjacent frames. However, when the parameter scale was enlarged to $C = 1024$, it shows more competitive results.

Second, the performance of the proposed method is substantially improved on both datasets when adding the depth-wise convolution to the merging module as shown in Figure 3(c), which indicates that the depth-wise convolution can effectively integrate the output features of the two branches in depth. Finally, adding SE-Block to the concatenation and depth-wise convolution, as shown in Figure 3(d), yields substantial performance improvement, achieving 0.899% and 6.922% EER on the VoxCeleb-E and CN-Celeb datasets, respectively, which is the state-of-the-art performance. This also shows that the SE operation is capable of effectively intensifying the features produced from the depth-wise convolution, thereby improving the efficiency of the feature fusion.

4. Conclusions

In this paper, we propose Branch-ECAPA-TDNN, a novel speaker embedding extractor for speaker verification. Branch-ECAPA-TDNN contains two parallel branches for extracting features with both a global range and various local ranges, where one branch uses the multi-head self-attention to capture long-range dependencies, and the other branch uses the SE-Res2Block module to extract local multi-scale characteristics. The local and global features rely on the merging module, which aims to enhance capability of the depth modeling, and further improve model performance. We investigated three merging methods, which are the concatenation-based, depth-wise-convolution-based and SE-based, respectively. The comprehensive experiments on the VoxCeleb and CN-Celeb datasets demonstrate the effectiveness of the proposed method.

5. Acknowledgements

This work was supported in part by the National Science Foundation of China (NSFC) under Grant 62176211, and in part by the Project of the Science, Technology, and Innovation Commission of Shenzhen Municipality, China under Grant JCYJ20210324143006016 and JSGG20210802152546026

6. References

[1] Z. Bai and X.-L. Zhang, "Speaker recognition based on deep learning: An overview," *Neural Networks*, vol. 140, pp. 65–99, 2021.

[2] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," *arXiv preprint arXiv:2003.11982*, 2020.

[3] 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.

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

[5] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *ICASSP*. IEEE, 2018, pp. 5329–5333.

[6] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[7] J. Zhou, T. Jiang, Z. Li, L. Li, and Q. Hong, "Deep speaker embedding extraction with channel-wise feature responses and additive supervision softmax loss function." in *Interspeech*, 2019, pp. 2883–2887.

[8] S. Ioffe, "Probabilistic linear discriminant analysis," in *ECCV*. Springer Berlin Heidelberg, 2006, pp. 531–542.

[9] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP 2019-2019 IEEE International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2019, pp. 5796–5800.

[10] 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.

[11] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "But system description to voxceleb speaker recognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[12] Z. Wang, K. Yao, X. Li, and S. Fang, "Multi-resolution multi-head attention in deep speaker embedding," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6464–6468.

[13] J. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *Interspeech 2018*, 2018.

[14] J.-w. Jung, H.-S. Heo, J.-h. Kim, H.-j. Shim, and H.-J. Yu, "Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification," *Proc. Interspeech 2019*, pp. 1268–1272, 2019.

[15] T. Zhou, Y. Zhao, and J. Wu, "Resnext and res2net structures for speaker verification," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 301–307.

[16] Y. Kwon, H.-S. Heo, B.-J. Lee, and J. S. Chung, "The ins and outs of speaker recognition: lessons from voxsrc 2020," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5809–5813.

[17] J. Thienpondt, B. Desplanques, and K. Demuynck, "Integrating frequency translational invariance in tdnns and frequency positional information in 2d resnets to enhance speaker verification," in *Interspeech2021*. ISCA, 2021, pp. 2302–2306.

[18] T. Liu, R. K. Das, K. A. Lee, and H. Li, "Mfa: Tdnn with multi-scale frequency-channel attention for text-independent speaker verification with short utterances," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7517–7521.

[19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[20] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 2, pp. 652–662, 2019.

[21] P. Safari, M. India, and J. Hernando, "Self-attention encoding and pooling for speaker recognition," *Proc. Interspeech 2020*, pp. 941–945, 2020.

[22] S. V. Katta, S. Umesh *et al.*, "S-vectors: Speaker embeddings based on transformer's encoder for text-independent speaker verification," *arXiv preprint arXiv:2008.04659*, 2020.

[23] Y. Zhang, Z. Lv, H. Wu, S. Zhang, P. Hu, Z. Wu, H.-y. Lee, and H. Meng, "Mfa-conformer: Multi-scale feature aggregation conformer for automatic speaker verification," *arXiv preprint arXiv:2203.15249*, 2022.

[24] K. Kim, F. Wu, Y. Peng, J. Pan, P. Sridhar, K. J. Han, and S. Watanabe, "E-branchformer: Branchformer with enhanced merging for speech recognition," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 84–91.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[26] C. Si, W. Yu, P. Zhou, Y. Zhou, X. Wang, and S. Yan, "Inception transformer," *arXiv preprint arXiv:2205.12956*, 2022.

[27] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[28] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[29] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[30] Y. Fan, J. Kang, L. Li, K. Li, H. Chen, S. Cheng, P. Zhang, Z. Zhou, Y. Cai, and D. Wang, "Cn-celeb: a challenging chinese speaker recognition dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7604–7608.

[31] L. Li, R. Liu, J. Kang, Y. Fan, H. Cui, Y. Cai, R. Vipperla, T. F. Zheng, and D. Wang, "Cn-celeb: multi-genre speaker recognition," *Speech Communication*, vol. 137, pp. 77–91, 2022.

[32] 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.

[33] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[34] E. A. Habets, "Room impulse response generator," *Technische Universiteit Eindhoven, Tech. Rep*, vol. 2, no. 2.4, p. 1, 2006.

[35] D. Garcia-Romero, A. McCree, D. Snyder, and G. Sell, "Jhu-hltcoe system for the voxsrc speaker recognition challenge," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7559–7563.

[36] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[37] H. Wang, C. Liang, S. Wang, Z. Chen, B. Zhang, X. Xiang, Y. Deng, and Y. Qian, "Wespeaker: A research and production oriented speaker embedding learning toolkit," *arXiv preprint arXiv:2210.17016*, 2022.

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