INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

ECAPA++: Fine-grained Deep Embedding Learning for TDNN Based Speaker Verification

Bei Liu, Yanmin Qian[†]

MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{beiliu, yanminqian}@sjtu.edu.cn

Abstract

In this paper, we aim to bridge the performance gap between TDNN and 2D CNN based speaker verification systems. Specifically, three types of architectural enhancements to ECAPA-TDNN are proposed: 1) follow depth-first design to significantly increase network depth while maintaining its complexity. 2) introduce recursive convolution to better capture fine-grained speaker information. 3) propose pyramid-based multi-path feature enhancement module to yield more discriminative speaker representation. Experiments on Voxceleb show that our final model, named ECAPA++, achieves **25%**, **23%** and **24%** relative improvements on Vox1-O, E and H respectively, while with **2.4x** fewer parameters and **2.3x** fewer FLOPs over the previous best TDNN-based system. Meanwhile, it is comparable to the state-of-the-art ResNet-based systems with higher computational efficiency. In addition, further performance gains can be achieved by fusing ECAPA++ and ResNet-based systems.

Index Terms: speaker verification, time-delay neural network, ECAPA, ResNet, system fusion

1. Introduction

The task of speaker verification (SV) is to determine whether testing and enrollment utterances belong to the same speaker or not. Traditionally, i-vector [1] combined with probabilistic linear discriminant analysis (PLDA) [2] dominates the speaker verification field. Since the advent of deep learning, neural networks have been highly applied in this task and become an indispensable part of the state-of-the-art systems gradually [3, 4, 5, 6, 7, 8, 9].

According to the network architecture, there are two main types of neural network based SV systems: time-delay neural network (TDNN) and 2-dimensional convolutional neural network (2D CNN). TDNN is characterized by the ability to efficiently model long temporal contexts between sequential data, which can be naturally applied to speech-related tasks. x-vector [3] firstly investigates the possibility of utilizing TDNN for speaker verification to replace traditional UBM method [1]. Subsequently, E-TDNN [10] and F-TDNN [11] are introduced to further improve the system performance by increasing context size. Recently, ECAPA-TDNN [5] and its variant [12] obtain the impressive results on Voxceleb dataset by enhancing x-vector with several architectural modifications. For 2D CNN-based SV systems, the winner of VoxSRC 2019 r-vector [4] proves that ResNet [13] with 2D convolution can also work excellently for SV task. DF-ResNet [8] proposes the depth-first version of r-vector to achieve a better trade-off on performance and complexity. Great speaker representation ability makes 2D CNN one of the most popular architectures in the SV field [7, 9, 14, 15]. Even though TDNN and 2D CNN are two mostly used speaker embedding extractors, the performance gap between them is becoming increasingly larger. In particular, this trend can be obviously reflected in recent speaker verification challenges. For example, the winners of VoxSRC 2021 [14] and CNSRC 2022 [15] both only adopt 2D CNN-based systems.

In this paper, we explore the question: is there the possibility of bridging the performance gap between TDNN and 2D CNN based SV systems? Our starting point is ECAPA-TDNN. Following the depth-first design rule, we first largely increase the network depth while maintaining its complexity. Then, in order to better capture fine-grained speaker information, a novel computational block named recursive convolution (RecConv) is proposed to replace the original Res2Conv. In addition, we introduce a pyramid-based multi-path feature enhancement module to fuse features across different layers, which yields more robust and discriminative speaker embeddings. Experiments on Voxceleb illustrate that compared with the previous best TDNN-based system, our resulting TDNN model, named ECAPA++, can achieve **25%**, **23%** and **24%** relative improvements on the three official trials respectively, while with **2.4x** fewer parameters and **2.3x** fewer FLOPs. Meanwhile, under higher computational efficiency, its performance is on par with the state-of-the-art ResNet-based systems. Plus, the complementarity analysis shows that further performance gains can be obtained by fusing ECAPA++ and ResNet-based systems.

2. Methodology

In this section, we describe the specific architectural designs to boost the performance of TDNN-based system based on ECAPA-TDNN [5]. Figure 1 schematically depicts the process of converting SE-Res2Block to SE-RecBlock. Table 1 presents the changes of parameter number, FLOPs and performance throughout the roadmap from ECAPA (C=512) to ECAPA++ (Small). Figure 3 is the overall architecture of our proposed ECAPA++.

2.1. Depth-First Design

Depth-first design rule is first proposed in [8]. Through largely deepening ResNet, great performance gains can be obtained without increasing the network complexity. Following the similar idea, we first significantly increase the depth of ECAPA (C=512) without adding extra computational overhead. The specific design choices are provided below.

channel downsampling: In the original ECAPA (C=512), there exist three successive SE-Res2Blocks each of which outputs the feature map with the same shape $C \times T$ where C is

[†] corresponding author

10.21437/Interspeech.2023-777

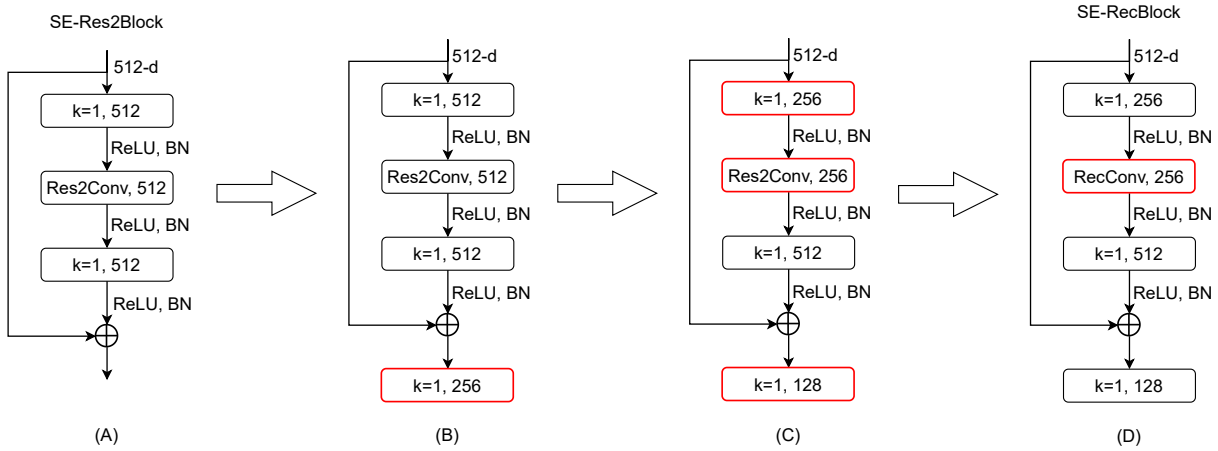

Figure 1: ***A-D:*** *The process of converting SE-Res2Block to SE-RecBlock. For simplicity, SE block is ignored in this figure.* ***A:*** *The original SE-Res2Block in ECAPA.* ***B:*** *Downsample the channel number by half after each block.* ***C:*** *Shrink the channel number of the first two layers in SE-Res2Block by half.* ***D:*** *Replace Res2Conv with RecConv.*

Table 1: *The roadmap from ECAPA (C=512) to ECAPA++ (Small) and the corresponding changes of parameter number, FLOPs and EER performance.*

System	# Params	FLOPs	Vox1-O
ECAPA (C=512)	6.2M	1.1G	1.01
channel downsampling	1.9M	0.3G	2.47
bottleneck-ify	1.4M	0.2G	2.65
increase block number	5.7M	1.0G	0.86
Res2Conv → RecConv	9.7M	1.9G	0.80
multi-path feature enhancement	14.7M	2.8G	0.76
ECAPA (C=1024)	14.7M	2.7G	0.87

set to 512. In order to reduce the parameter number, we follow the design paradigm of ResNet [13] and downsample the channel number C by a factor of 2 after each block as shown in Figure 1 (B). Unsurprisingly, this design choice significantly decreases the parameter number from 6.2M to 1.9M and FLOPs from 1.1G to 0.3G at the expense of the performance EER degrading to 2.47%, as Table 1 displays.

bottleneck-ify: The original SE-Res2Block consists of two 1D convolutional layers and one 1D dilated Res2Conv, as shown in Figure 1 (A). The number of channels in these three computational layers are all $C = 512$. Inspired by the channel expansion idea in bottleneck block of ResNet, we shrink the channel number C in the first two layers of SE-Res2Block by a factor of 2 (Figure 1 (C)). From Table 1, we can see that this change leads to a further decrease in parameters by 0.5M and FLOPs by 0.1G. Meanwhile, the EER temporarily reaches the highest point 2.65%.

increase block number: After the above steps, the parameter number and FLOPs are significantly reduced by 4.4x and 5.5x respectively. It is ready to deepen the model by increasing the block number. Following ResNet, we introduce the stage idea to build three computational stages in total each of which contains multiple blocks proposed in the previous step. In our design, the number of block in each stage is set to [8, 24, 8] respectively. This step significantly reduces the EER from 2.65% to 0.86%.

2.2. Recursive Convolution

As shown in Figure 2 (1), the original SE-Res2Block adopts the 1D convolutional version of Res2Net module [16]. For an input feature, Res2Conv firstly splits it equally along the channel dimension. Then the feature subsets are passed to 1D convolutional layer in a hierarchical way. We claim that the hierarchical structure can only model simple spatial interactions between the feature subsets. Inspired by [17], we introduce a novel computational block, namely recursive convolution (RecConv), to model higher-order spatial interactions in a recursive manner for better capturing fine-grained speaker information.

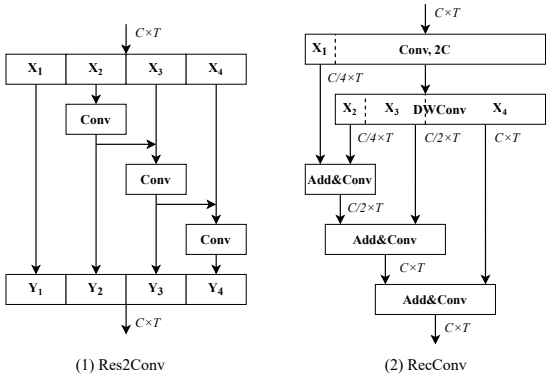

Figure 2: *Comparison between Res2Conv and the proposed RecConv (order $k = 3$)*

As Figure 2 (2) illustrates, RecConv firstly projects the channel number of input feature to $2C$ for channel mixing. Then a new dimension order (k) is introduced to control the level of spatial interactions among feature groups. For k-order interaction, the projected feature is split into $k + 1$ subsets, denoted as $\mathbf{x}_i$ where $i = 1, ..., k+1$, along the channel dimension according to the following rule.

$$[\mathbf{x}_1^{C/(2^{k-1})\times T}, \mathbf{x}_2^{C/(2^{k-1})\times T}, \cdots, \mathbf{x}_{k+1}^{C/(2^0)\times T}] \quad (1)$$

Then the feature subsets are processed in a recursive way.

$$\mathbf{y}_i = \begin{cases} \mathbf{x}_1 + \text{DWConv}(\mathbf{x}_2), & i = 2; \\ \text{Conv}(\mathbf{y}_{i-1}) + \text{DWConv}(\mathbf{x}_i), & 3 \le i \le k+1. \end{cases} \quad (2)$$

where DWConv is 1D depth-wise convolution. Conv means normal 1D convolution.

Finally, the last recursion step output $\mathbf{y}_{k+1}$ is fed into a Conv layer to obtain the final result of RecConv.

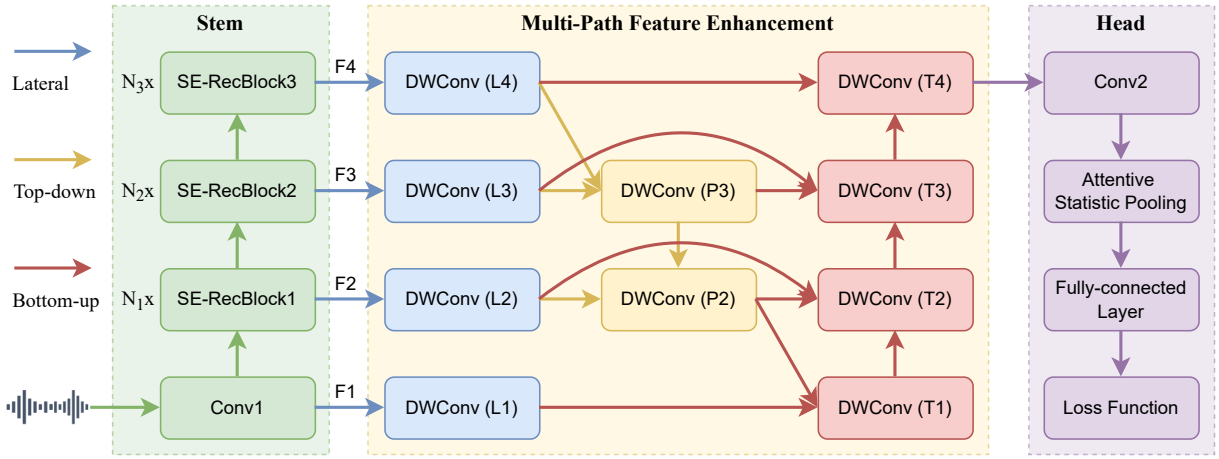

Figure 3: *The overall architecture of ECAPA++. It consists of three components.* ***Stem:*** *a stack of convolution layer and SE-RecBlocks which are divided into four stages.* ***Multi-Path Feature Enhancement:*** *contains three paths: lateral, top-down and bottom-up to aggregate features from the preceding stem.* ***Head:*** *includes pooling layer, fully-connected layer and loss function.*

Res2Conv $\rightarrow$ RecConv: By replacing Res2Conv with our proposed RecConv, this gives us a new block named SE-RecBlock (Figure 1 (D)). Accordingly, the EER is further decreased to 0.80%.

2.3. Pyramid-based Multi-path Feature Enhancement

The original ECAPA (C=512) exploits multi-layer feature aggregation by concatenating the output features of three SE-Res2Blocks together. In fact, evidence from [7] shows that direct concatenation is simple and non-learnable, lacking the ability of modeling complex interactions between features. Inspired by feature pyramid network used in object detection field [18], we propose pyramid-based multi-path feature enhancement module to aggregate features across different layers which consists of three paths: lateral, top-down and bottom-up, as shown in Figure 3.

The inputs are feature maps from different stages of the stem F_i where $i = 1, ..., 4$. Firstly, lateral convolutional operation is applied.

$$L_i = \text{DWConv}(F_i) \tag{3}$$

where DWConv is 1D depth-wise convolution. L_i denotes the i-th lateral path output.

Then, top-down layers are fed by lateral outputs and previous top-down features, resulting in a new intermediate feature map P_i.

$$P_i = \text{DWConv}(I^P_{i,1} \cdot L_i + I^P_{i,2} \cdot P_{i+1}) \tag{4}$$

$$I^P_{i,j} = \frac{e^{w^P_{i,j}}}{\sum_k e^{w^P_{i,k}}}, j, k = 1, 2 \tag{5}$$

where $w^P_{i,j}$ is learnable fusion weight parameter which can be normalized into $I^P_{i,j}$ through softmax function.

For bottom-up path, the i-th output T_i is rather computed by taking lateral, top-down and previous bottom-up features as inputs.

$$T_i = \text{DWConv}(I^T_{i,1} \cdot L_i + I^T_{i,2} \cdot P_i + I^T_{i,3} \cdot T_{i-1}) \tag{6}$$

$$I^T_{i,j} = \frac{e^{w^T_{i,j}}}{\sum_k e^{w^T_{i,k}}}, j, k = 1, 2, 3 \tag{7}$$

multi-path feature enhancement: As Table 1 presents, the performance can be improved to 0.76% after replacing the original multi-layer feature aggregation with our proposed module (Figure 3). This brings us to the final model named ECAPA++ (Small) which outperforms ECAPA (C=1024) by **13%** with similar parameters and FLOPs. The overall architecture is schematically depicted in Figure 3.

2.4. Construct A Family of ECAPA++

Based on the above ECAPA++ (Small), we build another variant which only differs in the layer number. The specific configurations of channel number C and block number B for SE-RecBlock in Figure 3 are listed below:

- ECAPA++ (Small): C=[256, 128, 64], B=[8, 24, 8]
- ECAPA++ (Big): C=[256, 128, 64], B=[16, 48, 16]

3. Experimental Setup

3.1. Datasets

Our experiments are conducted on the Voxceleb1&2 [20, 21] datasets. The development set of Voxceleb2 is used as training data, and the testing data is Voxceleb1. Performance is measured on the three official trial lists. Plus, data augmentation techniques are utilized to improve the robustness of systems, including online data augmentation [22] with MUSAN [23] and RIR dataset [24], specaugment [25] and speed perturb [26] with 0.9 and 1.1 times speed changes.

3.2. Implementation Details

All the systems are implemented using PyTorch framework. For the experiments, 80-dimensional Fbank features with 25ms windows and 10ms shift are extracted as input features. We randomly chunk a 200-frame segment from each utterance during the training process. AAM-softmax [27] with a margin of 0.2 and a scale of 32 is used as loss function. Models are optimized using AdamW [28] with a weight decay of 0.05.

3.3. Evaluation Metrics

During testing, we use cosine distance as the scoring criterion. Then, all the scores are normalized using adaptive score normalization (AS-Norm) [29] where the size of the imposter cohort is set to 600. Performance is measured in terms of the equal error rate (EER) and the minimum detection cost function (MinDCF) with the settings of $P_{target} = 0.01$ and $C_{FA} = C_{Miss} = 1$.

Table 2: *EER and MinDCF results of previous systems and our proposed ECAPA++ on the Voxceleb1 dataset.*

System	Architecture	# Params	FLOPs	Voxceleb-O		Voxceleb-E		Voxceleb-H	
				EER(%)	MinDCF	EER(%)	MinDCF	EER(%)	MinDCF
ResNet101 [8]	ResNet	15.9M	10.1G	0.62	0.0633	0.80	0.0880	1.48	0.1431
DF-ResNet179 [8]		9.8M	8.6G	0.62	0.0611	0.80	0.0899	1.51	0.1483
ECAPA (C=1024) [5]	TDNN	14.7M	2.7G	0.87	0.1066	1.12	0.1318	2.12	0.2101
ECAPA (C=2048) [19]		56.2M	10.7G	0.86	0.0960	1.08	0.1223	2.01	0.2004
ECAPA++ (Small)	TDNN	14.7M	2.8G	0.76	0.0960	0.96	0.1144	1.72	0.1671
ECAPA++ (Big)	(ours)	23.9M	4.6G	0.65	0.0796	0.84	0.0981	1.54	0.1536

4. Results and Analysis

4.1. Main Results

The performance overview of the state-of-the-art ResNet-based, TDNN-based and our proposed ECAPA++ systems are presented in Table 2. Meanwhile, parameter number and FLOPs are provided for the detailed comparison of model complexity.

Firstly, it can be clearly seen that there exists a significant performance gap between TDNN and ResNet based systems. This reveals that 2D CNN inherently exhibits much superior speaker representation ability over previous TDNN architectures, which urges the necessity of introducing architectural enhancements to TDNN model.

For our proposed ECAPA++, there exists a new dimension called order (k) in SE-RecBlock. In the experiments, we set $k = 5, 4, 3$ for SE-Recblock1, 2, 3 respectively in Figure 3 by default. In addition, the channel number C in multi-path feature enhancement module is set to 512. From Table 2, we can see that the proposed ECAPA++ (Small) obtains average relative improvements in EER by 16% and in MinDCF by 15% respectively over ECAPA (C=512) with similarly-sized parameters and FLOPs. Compared with previous best TDNN model ECAPA (C=2048), our ECAPA++ (Big) achieves a new state-of-the-art performance, which results in relative improvements in EER by 25%, 23%, 24% and in MinDCF by 18%, 20%, 24% in the three official trials, together with 2.4x fewer parameters and 2.3x fewer FLOPs. This reveals that our architectural designs introduced in section 2 are not only powerful and effective, which can yield more robust and discriminative speaker representation, but also computationally efficient. Moreover, ECAPA++ (Big) is on par with ResNet101 and DF-ResNet179 which illustrates that the performance gap is successfully bridged. Another advantage of ECAPA++ (Big) over ResNet-based systems is computational efficiency, and FLOPs are reduced to a half of ResNet, as Table 2 shows.

4.2. Analysis of the Complementarity between Systems

System fusion is a widely-used technique in various speaker verification challenges [30, 14, 15, 31]. Prior studies show that significant performance gains can be achieved by fusing TDNN and ResNet-based systems [30]. However, TDNN-based systems are gradually discarded in recent challenges due to its poor performance [14, 15, 31]. For example, fusing the SOTA ResNet-based SV systems with ECAPA-TDNN will not yield extra performance improvements. In this section, we investigate the embedding complementarity between our proposed ECAPA++ and ResNet-based systems through score fusion.

From Table 3, we can see that the fusion between ResNet101/DF-ResNet179 and the previous best TDNN system ECAPA (C=2048) can not result in further performance gains. This phenomenon reveals that ECAPA (C=2048) can not provide complementary information to ResNet101/DF-ResNet179 due to the large performance gap. In contrast, our proposed ECAPA++ successfully bridge the performance gap between TDNN and ResNet-based systems and further improvements can be obtained by fusing ECAPA++ (Big) and ResNet101/DF-ResNet179. This confirms the complementarity between our proposed ECAPA++ and the SOTA ResNet-based systems, indicating the potential application of ECAPA++ in system fusion for speaker verification challenges.

Table 3: *Results of system fusion between TDNN and ResNet101/DF-ResNet179. System fusion is based on score weighted summation.*

System 1	System 2	EER (%)		
		Vox1-O	Vox1-E	Vox1-H
ResNet101	–	0.62	0.80	1.48
	ECAPA (C=2048)	0.62	0.79	1.49
	ECAPA++ (Big)	**0.55**	**0.71**	**1.32**
DF-ResNet179	–	0.62	0.80	1.51
	ECAPA (C=2048)	0.61	0.79	1.50
	ECAPA++ (Big)	**0.54**	**0.70**	**1.30**

5. Conclusions

In this paper, we aim to bridge the performance gap between TDNN and 2D CNN based speaker verification systems. Specifically, three types of architectural enhancements to ECAPA-TDNN are proposed including depth-first design, recursive convolution and propose pyramid-based multi-path feature enhancement module. Experiments on Voxceleb show that our final model, named ECAPA++, achieves **25%**, **23%** and **24%** relative improvements on Vox1-O, E and H respectively, while with **2.4x** fewer parameters and **2.3x** fewer FLOPs over the previous best TDNN-based system. Meanwhile, it is comparable to the state-of-the-art ResNet-based systems with higher computational efficiency. Furthermore, the complementarity analysis illustrates that extra performance gains can be obtained by fusing ECAPA++ and ResNet-based systems.

6. Acknowledgement

This work was supported in part by China NSFC projects under Grants 62122050 and 62071288, and in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

7. 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, 2011.

[2] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision (ECCV)*, 2006, pp. 531–542.

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

[4] 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.

[5] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020, pp. 3830–3834.

[6] B. Liu, H. Wang, Z. Chen, S. Wang, and Y. Qian, "Self-knowledge distillation via feature enhancement for speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7542–7546.

[7] B. Liu, Z. Chen, and Y. Qian, "Attentive feature fusion for robust speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2022, pp. 286–290.

[8] B. Liu, Z. Chen, S. Wang, H. Wang, B. Han, and Y. Qian, "Df-resnet: boosting speaker verification performance with depth-first design," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2022, pp. 296–300.

[9] B. Liu, Z. Chen, and Y. Qian, "Dual path embedding learning for speaker verification with triplet attention," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2022, pp. 291–295.

[10] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5796–5800.

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

[12] J. Thienpondt, B. Desplanques, and K. Demuynck, "The idlab voxsrc-20 submission: Large margin fine-tuning and quality-aware score calibration in dnn based speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5814–5818.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[14] M. Zhao, Y. Ma, M. Liu, and M. Xu, "The speakin system for voxceleb speaker recognition challange 2021," *arXiv preprint arXiv:2109.01989*, 2021.

[15] Z. Chen, B. Liu, B. Han, L. Zhang, and Y. Qian, "The sjtu x-lance lab system for cnsrc 2022," *arXiv preprint arXiv:2206.11699*, 2022.

[16] S. Gao, M. Cheng, K. Zhao, X. Zhang, M. 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, 2021.

[17] Y. Rao, W. Zhao, Y. Tang, J. Zhou, S. N. Lim, and J. Lu, "Hornet: Efficient high-order spatial interactions with recursive gated convolutions," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, pp. 10 353–10 366.

[18] M. Tan, R. Pang, and V. Quoc, "Efficientdet: scalable and efficient object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 781–10 790.

[19] J. Thienpondt, B. Desplanques, and K. Demuynck, "The idlab voxsrc-20 submission: Large margin fine-tuning and quality-aware score calibration in dnn based speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5799–5803.

[20] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 2616–2620.

[21] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 1086–1090.

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

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

[24] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[25] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: a simple data augmentation method for automatic speech recognition," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2613–2617.

[26] W. Wang, D. Cai, X. Qin, and M. Li, "The dku-dukeece systems for voxceleb speaker recognition challenge 2020," *arXiv preprint arXiv:2010.12731*, 2020.

[27] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2873–2877.

[28] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, 2019.

[29] Z. N. Karam, W. M. Campbell, and N. Dehak, "Towards reduced false-alarms using cohorts," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4512–4515.

[30] J. Thienpondt, B. Desplanques, and K. Demuynck, "The idlab voxceleb speaker recognition challenge 2020 system description," *arXiv preprint arXiv:2010.12468*, 2020.

[31] Z. Chen, B. Han, X. Xiang, H. Huang, B. Liu, and Y. Qian, "Sjtu-aispeech system for voxceleb speaker recognition challenge 2022," *arXiv preprint arXiv:2209.09076*, 2022.