INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

CAM++: A Fast and Efficient Network for Speaker Verification Using Context-Aware Masking

Hui Wang, Siqi Zheng, Yafeng Chen, Luyao Cheng, Qian Chen

Speech Lab, Alibaba Group

{tongmu.wh, zsq174630}@alibaba-inc.com

Abstract

Time delay neural network (TDNN) has been proven to be efficient for speaker verification. One of its successful variants, ECAPA-TDNN, achieved state-of-the-art performance at the cost of much higher computational complexity and slower inference speed. This makes it inadequate for scenarios with demanding inference rate and limited computational resources. We are thus interested in finding an architecture that can achieve the performance of ECAPA-TDNN and the efficiency of vanilla TDNN. In this paper, we propose an efficient network based on context-aware masking, namely CAM++, which uses densely connected time delay neural network (D-TDNN) as backbone and adopts a novel multi-granularity pooling to capture contextual information at different levels. Extensive experiments on two public benchmarks, VoxCeleb and CN-Celeb, demonstrate that the proposed architecture outperforms other mainstream speaker verification systems with lower computational cost and faster inference speed.

Index Terms: speaker verification, densely connected time delay neural network, context-aware masking, computational complexity

1. Introduction

Speaker verification (SV) is the task of automatically verifying whether an utterance is pronounced by a hypothesized speaker based on the voice characteristic [1]. Typically, a speaker verification system consists of two main components - an embedding extractor which transforms an utterance of random length into a fixed-dimensional speaker embedding, and a back-end model that calculates the similarity score between the embeddings [2, 3].

Over past few years, speaker verification systems based on deep learning methods [2, 4, 5, 6, 7] have achieved remarkable improvements. One of the most popular systems is x-vector, which adopts time delay neural network (TDNN) as backbone. TDNN takes one-dimensional convolution along the time axis to capture local temporal context information. Following the successful application of x-vector, several modifications are proposed to enhance robustness of the networks. ECAPA-TDNN [4] unifies one-dimensional Res2Block with squeeze-excitation [8] and expands the temporal context of each layer, achieving significant improvement. At the same time, the topology of x-vector is improved by incorporating elements of ResNet [9] which uses a two-dimensional convolutional neural network (CNN) with convolutions in both time and frequency axes. Equiped with residual connection, ResNet-based systems [10, 11] have achieved outstanding results. However, these networks tend to require a large number of parameters and computations to achieve optimal performance. In real-world applications, accuracy and efficiency are equally important. It is of sufficient interest and challenge to find a speaker embedding extracting network that simultaneously improves the performance, computation complexity, and inference speed.

Recently, [5] proposes a TDNN-based architecture, called densely connected time delay neural network (D-TDNN), by adopting bottleneck layers and dense connectivity. It obtains better accuracy with fewer parameters compared to vanilla TDNN. Later, in [6], a context-aware masking (CAM) module is proposed to make the D-TDNN focus on the speaker of interest and "blur" unrelated noise, while requiring only a little computation cost. Despite of significant improvements on accuracy, there still exists a large performance gap compared to other state-of-the-art speaker models [4].

In this paper, we propose CAM++, an efficient and accurate network for speaker embedding learning that utilizes D-TDNN as a backbone, as shown in Figure 1. We have adopted multiple methodologies to enhance the CAM module and D-TDNN architecture. Firstly, we design a lighter CAM module and insert it into each D-TDNN layer to place more focus on the speaker characteristics of interest. Multi-granularity pooling is an essential component of the CAM module, built to capture contextual information at both global and segment levels. The previous study in [12] showed that multi-granularity pooling achieves comparable performance with much higher efficiency, when compared to a transformer structure. Secondly, we adopt a narrower network with fewer filters in each D-TDNN layer, significantly increasing the network depth compared to vanilla D-TDNN [5]. This is motivated by [11], which observed that deeper layers can bring more improvements than wider channels for speaker verification. Finally, we incorporate a two-dimensional convolution module as a front-end to enhance the D-TDNN network's ability to be invariant to frequency shifts in the input features. A hybrid architecture of TDNN and CNN has been shown to yield further improvements [13, 14]. We evaluate the proposed architecture on two public benchmarks, VoxCeleb [15] and CN-Celeb [16, 17]. The results show that our method obtains 0.73% and 6.78% EER in VoxCeleb-O and CN-Celeb test sets. Furthermore, our architecture has lower computation complexity and faster inference speed than popular ECAPA-TDNN and ResNet34 systems.

2. System description

2.1. Overview

The overall framework of the proposed CAM++ architecture is illustrated in Figure 1. The architecture mainly consists of two components: the front-end convolution module (FCM) and the D-TDNN backbone. The FCM consists of multiple blocks of two-dimensional convolution with residual connections, which

10.21437/Interspeech.2023-1513

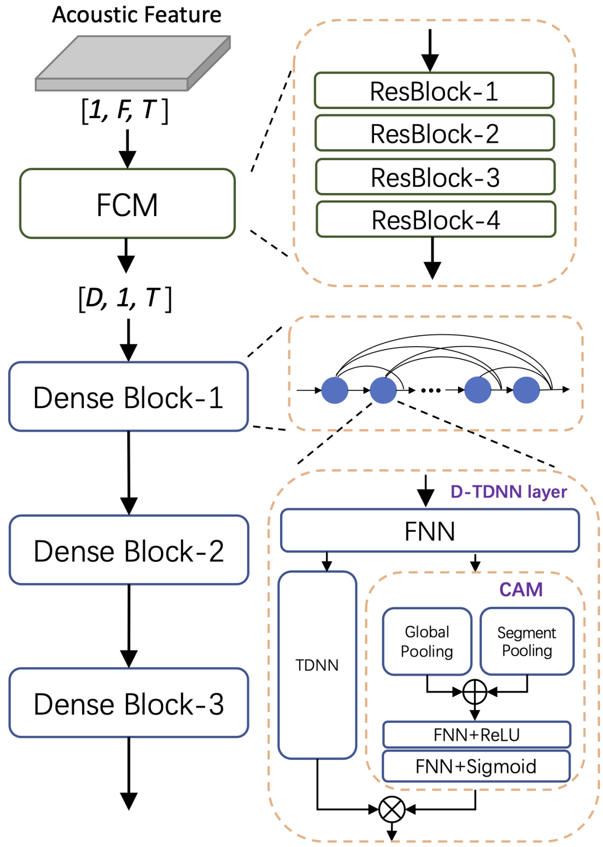

Figure 1: *Overview of the proposed CAM++ architecture. It comprises convolution modules as the front-end and D-TDNN as the backbone. An improved context-aware making is built into each D-TDNN layer, which includes multi-granularity pooling to capture speaker characteristics.*

encode acoustic features in the time-frequency domain to exploit high-resolution time-frequency details. The resulting feature map is subsequently flattened along the channel and frequency dimensions and used as input for the D-TDNN. The D-TDNN backbone comprises three blocks, each containing a sequence of D-TDNN layers. In each D-TDNN layer, we build an improved CAM module that assigns different attention weights to the output feature of the inner TDNN layer. The multi-granularity pooling incorporates global average pooling and segment average pooling to effectively aggregate contextual information across different levels. With dense connections, the masked output is concatenated with all preceding layers and serves as the input for the next layer.

2.2. D-TDNN backbone

TDNN uses a dilated one-dimensional convolution structure along the time axis as its backbone, which was first adopted by x-vector [2]. Due to its success, TDNN has been widely used in speaker verification tasks. An improved version, D-TDNN, was recently proposed in [5] as an efficient TDNN-based speaker embedding model. Similar to DenseNet [18], it adopts dense connectivity, which involves direct connections among all layers in a feed-forward manner. D-TDNN is parameter-efficient and achieves better results while requiring fewer parameters than vanilla TDNN. Hence, we adopt D-TDNN as the backbone of our network.

Specifically, the basic unit of D-TDNN consists of a feed-forward neural network (FNN) and a TDNN layer. A direct connection is applied between the input of two consecutive D-TDNN layers. The formulation of the l-th D-TDNN layer is:

$$\boldsymbol{S}^l = \mathcal{H}_l([\boldsymbol{S}^0, \boldsymbol{S}^1, \cdots, \boldsymbol{S}^{l-1}]) \tag{1}$$

where $\boldsymbol{S}^0$ is the input of the D-TDNN block, $\boldsymbol{S}^l$ is the output of the the l-th D-TDNN layer, $\mathcal{H}_l$ denotes the non-linear transformation of the l-th D-TDNN layer.

Although D-TDNN has demonstrated remarkable improvement in comparison to vanilla TDNN, there remains a considerable gap between it and state-of-the-art speaker embedding models like ECAPA-TDNN and ResNet34. We redesign the D-TDNN to further push its limits and achieve better results. In [11], it is revealed that depth of the network plays a critical role in the performance of speaker verification, and increasing the depth of the speaker embedding model tends to yield more improvement than widening it. Hence, we significantly increase the depth of the D-TDNN network while reducing the channel size of filters in each layer to control the network's complexity. Specifically, the vanilla D-TDNN has two blocks, each containing 6 and 12 D-TDNN layers, respectively. We add an additional block at the end and expand the number of layers per block to 12, 24 and 16. To reduce the network's complexity, we adopt narrower D-TDNN layers in each block, that is, reducing the original growth rate k from 64 to 32. Additionally, we adopted an input TDNN layer with 1/2 subsampling rate before the D-TDNN backbone to accelerate computation. In Section 3.3, the experimental results will indicate that these effective modifications significantly improve the performance of speaker verification.

2.3. Context-aware masking

Attention mechanism has been widely adopted in speaker verification. Squeeze-excitation (SE) [8] squeezes global spatial information into a channel descriptor to model channel interdependencies and recalibrate filter responses. Meanwhile, soft self-attention is utilized to calculate the weighted statistics for the improvement of temporal pooling techniques [19, 20, 21].

An attention-based context-aware masking (CAM) module was recently proposed in [6] to focus on the speaker of interest and blur unrelated noise, resulting a significant improvement in the performance of D-TDNN. CAM performs feature map masking using an auxiliary utterance-level embedding obtained from global statistic pooling. However, in [6], CAM is only applied at the transition layer after each D-TDNN block, and a limited number of CAM modules may be insufficient for extracting critical information effectively. To address this, we propose a lighter CAM and insert it into each D-TDNN layer to capture more speaker characteristic of interest.

As shown in Figure 1, we denote the output hidden feature from the head FNN in the D-TDNN block as $\boldsymbol{X}$. Firstly, $\boldsymbol{X}$ is input into the TDNN layer to extract local temporal feature $\boldsymbol{F}$:

$$\boldsymbol{F} = \mathcal{F}(\boldsymbol{X}) \tag{2}$$

where $\mathcal{F}(\cdot)$ denotes the transformation of the TDNN layer. $\mathcal{F}(\cdot)$ only focuses on local receptive field and $\boldsymbol{F}$ may be suboptimal. Therefore, a ratio mask $\boldsymbol{M}$ is predicted based on an extracted contextual embedding $\boldsymbol{e}$, and is expected to contain both speaker of interest and noise characteristic.

$$\boldsymbol{M}_{*t} = \sigma(\boldsymbol{W}_2\delta(\boldsymbol{W}_1\boldsymbol{e} + \boldsymbol{b}_1) + \boldsymbol{b}_2) \tag{3}$$

where $\sigma(\cdot)$ denotes the Sigmoid function, $\delta(\cdot)$ denotes the ReLU function, and $\boldsymbol{M}_{*t}$ denotes the t-th frame of $\boldsymbol{M}$.

In [6], a global statistic pooling is used to generate the contextual embedding $\boldsymbol{e}$. It is known that speech signals have typical hierarchical structure and exhibit dynamic changes in characteristic between different subsegments. A unique speaking manner of the target speaker may exist within a certain segment. Simply using a single embedding from global pooling may result in loss of precise local contextual information, leading to a suboptimal masking. Therefore, it is beneficial to extend the single global pooling to multi-granularity pooling. This enables the network to capture more contextual information at different levels, generating a more accurate mask. Specifically, a global average pooling is used to extract contextual information at global level:

$$\boldsymbol{e}_g = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{X}_{*t} \tag{4}$$

Simultaneously, a segment average pooling is used to extract contextual information at segment level:

$$\boldsymbol{e}_s^k = \frac{1}{s_{k+1}-s_k}\sum_{t=s_k}^{s_{k+1}-1}\boldsymbol{X}_{*t} \tag{5}$$

Where s_k is the starting frames of k-th segment of feature $\boldsymbol{X}$. In the experiments, we segment the frame-level feature $\boldsymbol{X}$ into consecutive fixed-length 100-frame segments and apply segment average pooling to each.

Subsequently, contextual embeddings of different level , $\boldsymbol{e}_g$ and $\boldsymbol{e}_s$, are aggregated to predict the context-aware mask $\boldsymbol{M}$. The Equation 3 can be rewrote to:

$$\begin{aligned}\boldsymbol{M}_{*t}^k =& \sigma(\boldsymbol{W}_2\delta(\boldsymbol{W}_1(\boldsymbol{e}_g+\boldsymbol{e}_s^k)+\boldsymbol{b}_1)+\boldsymbol{b}_2),\\ & s_k \leqslant t < s_{k+1}\end{aligned} \tag{6}$$

Finally, predicted $\boldsymbol{M}$ is used to calibrate the representation and produce the refined representation $\tilde{\boldsymbol{F}}$.

$$\tilde{\boldsymbol{F}} = \mathcal{F}(\boldsymbol{X}) \odot \boldsymbol{M} \tag{7}$$

Where $\odot$ denotes the element-wise multiplication. Equation 6 has a simpler form and fewer trainable parameters compared to [6]. We insert this efficient context-aware masking into each D-TDNN layer to enhance the representational power of basic layers throughout the network.

2.4. Front-end convolution module

TDNN-based networks perform one-dimension convolution along the time axis, using kernels that cover the complete frequency range of the input features. It is more difficult to capture speaker characteristics occurring at certain local frequency regions compared to two-dimensional convolutional network [13]. Generally, plenty of filters are required to model the complex details in the full frequency region. For examples, ECAPA-TDNN has a maximum of 1024 channels in the convolutional layers to achieve optimal performance. In Section 2.2, we use narrower layers in each D-TDNN block to control the size of parameters. This may result in a reduced ability to find the specific frequency pattern in some local regions. It is necessary to enhance the robustness of D-TDNN to small and reasonable shifts in the time-frequency domain and compensate for realistic intra-speaker pronunciation variability. Motivated by [13, 14], we equip the D-TDNN network with a two-dimensional front-end convolution module (FCM). Inspired by the success of ResNet-based architectures in speaker verification, we decide to incorporate 4 residual blocks in the FCM stem, as illustrated in Figure 1. The number of channels is set to 32 for all residual blocks. We use a stride of 2 in the frequency dimension in the last three blocks, resulting in an 8x downsampling in the frequency domain. The output feature map of FCM is subsequently flattened along the channel and frequency dimension and used as input for the D-TDNN backbone.

3. Experiments

3.1. Dataset

We conduct experiments on two public speaker verification benchmarks, VoxCeleb [15] and CN-Celeb [16, 17], to evaluate the effectiveness of the proposed methods. For VoxCeleb, we use the development set of VoxCeleb2 for training, which contains 5,994 speakers. The evaluation set is constructed from three cleaned version test trials, VoxCeleb1-O, VoxCeleb1-E and VoxCeleb1-H. The last two tasks have more trial pairs. For CN-Celeb, the development sets of CN-Celeb1 and CN-Celeb2 are used for training, which contain 2785 speakers. In the data preprocessing of the training data, we concatenate short utterances to ensure that they are no less than 6s. There exists multiple utterances for each enrollment speaker in CN-Celeb test set. We choose to average all the embeddings which belong to the same enrollment speaker to get final speaker embedding for evaluation.

3.2. Experimental setup

For all experiments, we use 80-dimensional Fbank features extracted over a 25 ms long window for every 10 ms as input. We apply speed perturbation augmentation by randomly sampling a ratio from $\{0.9, 1.0, 1.1\}$. The processed audio is considered to be from a new speaker [22]. In addition, two popular data augmentations are adopted during training, simulating reverberation using the RIR dataset [23], adding noise using the MUSAN dataset [24].

Angular additive margin softmax (AAM-Softmax) loss [25] is used for all experiments. The margin and scaling factors of AAM-Softmax loss are set to 0.2 and 32 respectively. During training, we adopt stochastic gradient descent (SGD) optimizer with a cosine annealing scheduler and a linear warm-up scheduler, where the learning rate is varied between 0.1 and 1e-4. The momentum is 0.9, and the weight decay is 1e-4. 3s-long samples are randomly cropped from each audio to construct the training minibatches.

We use cosine similarity scoring for evaluation, without applying score normalization in the back-end. We adopt two commonly used metrics in speaker verification tasks, equal error rate (EER) and the minimum detection cost function (MinDCF) with 0.01 target probability.

3.3. Results on VoxCeleb and CN-Celeb

The performance overview of all methods is presented in Table 1. For fair comparison, we re-implement several baseline models under the same experimental setup described in Section 3.2, including TDNN [2], D-TDNN [5], ECAPA-TDNN [4] and ResNet34 [10]. The ResNet34 model contains four residual blocks with different channel sizes, [64, 128, 256, 512], in each block. The ECAPA-TDNN model with 1024 channels is built according to [4].

It can be found in Table 1 that, as an improved variant

Table 1: *Performance comparison of different network architectures on the VoxCeleb1 and CN-Celeb test sets. Data augmentation strategy and experimental setup are kept consistent throughout all experiments.*

Architecture	Params(M)	VoxCeleb1-O EER(%)/MinDCF	VoxCeleb1-E EER(%)/MinDCF	VoxCeleb1-H EER(%)/MinDCF	CN-Celeb Test EER(%)/MinDCF
TDNN	4.62	2.31/0.3223	2.37/0.2732	4.25/0.3931	9.86/0.6199
ECAPA-TDNN	14.66	0.89/0.0921	1.07/0.1185	1.98/0.1956	7.45/0.4127
ResNet34	6.70	0.97/0.0877	1.03/0.1133	1.88/0.1778	6.97/0.3859
D-TDNN	2.85	1.55/0.1656	1.63/0.1748	2.86/0.2571	8.41/0.4683
D-TDNN-L	6.40	1.19/0.1179	1.21/0.1287	2.22/0.2047	7.82/0.4336
CAM++	7.18	**0.73/0.0911**	**0.89/0.0995**	**1.76/0.1729**	**6.78/0.3830**
-w/o Masking	6.64	0.93/0.1022	1.03/0.1144	1.86/0.1762	7.16/0.3947
-w/o FCM	6.94	0.98/0.1127	1.01/0.1175	2.03/0.2006	7.17/0.4011

Table 2: *Performance comparison of multiple key components of CAM++. GP represents masking with only global pooling and SP denotes segment pooling.*

Method	Params(M)	CN-Celeb Test	
		EER(%)	MinDCF
D-TDNN	2.85	8.41	0.4683
CAM [6]	4.10	7.80	0.4431
GP	3.07	7.78	0.4321
GP+SP	3.07	**7.59**	**0.4209**

of TDNN, ECAPA-TDNN achieves impressive improvement in EER and MinDCF but requires large amounts of parameters. Using dense connection, D-TDNN outperforms TDNN with fewer parameters. Compared to the standard D-TDNN, it can be found that deeper D-TDNN-L proposed in Section 2.2 achieves significant performance improvement, thanks to increased parameters and effective modifications. However, there is still a large performance gap compared to ECAPA-TDNN or ResNet34. When we equip the D-TDNN-L backbone with CAM with multi-granularity pooling and FCM, CAM++ consistently performs better than the ECAPA-TDNN and ResNet34 baselines. In particular, CAM++ has relative 51% fewer parameters and 18% lower EER than ECAPA-TDNN in VoxCeleb-O.

Next, we remove individual components to explore the contribution of each to the performance improvements. It can be observed that CAM with multi-granularity pooling improves the EER in VoxCeleb-O and CN-Celeb test sets by 21% and 5%, respectively. This confirms the benefit of aggregating contextual vectors at different levels to perform attention masking. Removing FCM leads to a obvious increase in EER and MinDCF in all test sets. This phenomenon indicates that stronger speaker embeddings can be obtained from a hybrid of two-dimensional convolution and TDNN-based network.

3.4. Impacts of multi-granularity pooling

We further evaluate the effectiveness of the improved CAM with multi-granularity pooling. Additional experimental results on the CN-Celeb test set are presented in Table 2. We use D-TDNN [5] as the baseline. We re-implement the CAM proposed in [6] on CN-Celeb, and find it decrease the EER by 7% relatively but with a 44% increase in parameters. Next, We apply the improved CAM proposed in Section 2.3 to D-TDNN only with global average pooling (GP), which results in similar improvement in EER with only an 8% increase in parameters, demonstrating better parameters efficiency. We then apply segment average pooling (SP) and fuse it with GP, observing performance gains without introducing additional parameters. These results indicate the importance of local segment contextual information in performing more accurate masking.

Table 3: *The number of parameters, floating-point operations (FLOPs) and real-time factor (RTF) of different models. RTF was evaluated on CPU under single-thread condition.*

Model	Params(M)	FLOPs(G)	RTF
ECAPA-TDNN	14.66	3.96	0.033
ResNet34	6.70	6.84	0.032
CAM++	7.18	**1.72**	**0.013**

3.5. Complexity analysis

In this section, we compare the complexity of ECAPA-TDNN, ResNet34 and CAM++ models, including the number of parameters, floating-point operations (FLOPs) and real-time factor (RTF), as shown in Table 3. RTF was evaluated on the CPU device under single-thread condition. When comparing CAM++ with ResNet34, CAM++ has slightly more parameters but significant fewer FLOPs. At the same time, CAM++ has half the parameters and FLOPs of ECAPA-TDNN. It is worth noting that CAM++ achieves more than twice the inference speed of both ResNet34 and ECAPA-TDNN. Although ResNet34 and ECAPA-TDNN have a similar RTF, they have different FLOPs. This is likely due to increased memory access resulting from higher parameter data dependencies, which leads to increased computation time.

4. Conclusion

This paper proposed CAM++, an efficient speaker embedding model for speaker verification. Our novel context-aware masking method aimed to focus on the speaker of interest and improved the quality of features, while multi-granularity pooling fused different levels of contextual information to generate accurate attention weights. We conducted comprehensive experiments on two public benchmarks, VoxCeleb and CN-Celeb. The results demonstrated that CAM++ achieved superior performance with lower computational complexity and faster inference speed than popular ECAPA-TDNN and ResNet34 systems.

5. References

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

[2] 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.

[3] S. Zheng, G. Liu, H. Suo, and Y. Lei, "Autoencoder-based semi-supervised curriculum learning for out-of-domain speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 4360–4364.

[4] 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.

[5] Y.-Q. Yu and W.-J. Li, "Densely connected time delay neural network for speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020, pp. 921–925.

[6] Y.-Q. Yu, S. Zheng, H. Suo, Y. Lei, and W.-J. Li, "Cam: Context-aware masking for robust speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6703–6707.

[7] S. Zheng, Y. Lei, and H. Suo, "Phonetically-aware coupled network for short duration text-independent speaker verification," in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association*. ISCA, 2020, pp. 926–930.

[8] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[9] 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.

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

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

[12] C. Tan, Q. Chen, W. Wang, Q. Zhang, S. Zheng, and Z. Ling, "Ponet: Pooling network for efficient token mixing in long sequences," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

[13] 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 *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2021, pp. 2302–2306.

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

[15] A. Nagrani, J. S. Chung, W. Xie, , and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech and Language*, vol. 60, 2020.

[16] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7604–7608.

[17] 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*, 2022.

[18] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, p. 2261–2269.

[19] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 2252–2256.

[20] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text independent speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 3573–3577.

[21] M. Indiay, P. Safariy, and J. Hernando, "Self multi-head attention for speaker recognition," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 4305–4309.

[22] Z. Chen, B. Han, X. Xiang, H. Huang, B. Liu, and Y. Qian, "Build a sre challenge system: Lessons from voxsrc 2022 and cnsrc 2022," 2022, arXiv:2211.00815v1.

[23] 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.

[24] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[25] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arc-face: Additive angular margin loss for deep face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4690–4699.