Interspeech 2022
18-22 September 2022, Incheon, Korea

CTFALite: Lightweight Channel-specific Temporal and Frequency Attention Mechanism for Enhancing the Speaker Embedding Extractor

Yuheng Wei[1], *Junzhao Du*[1*], *Hui Liu*[1*], *Qian Wang*[1]

[1]School of Computer Science and Technology, Xidian University

weiyuheng@stu.xidian.edu.cn, dujz@xidian.edu.cn, liuhui@xidian.edu.cn,
qwang_97@stu.xidian.edu.cn

Abstract

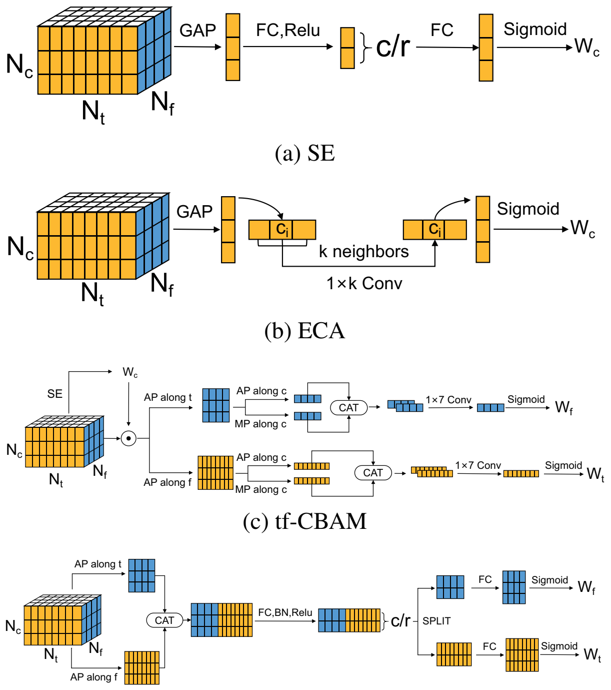
Attention mechanism provides an effective and plug-and-play feature enhancement module for speaker embedding extractors. Attention-based pooling layers have been widely used to aggregate a sequence of frame-level feature vectors into an utterance-level speaker embedding. Besides, convolution attention mechanisms are introduced into convolution blocks to improve the sensibility of speaker embedding extractors to those features with more discriminative speaker characteristics. However, it is still a challenging problem to make a good trade off between performance and model complexity for convolution attention models, especially for speaker recognition systems on low-resource edge computing nodes (smartphone, embedded devices, etc.). In this paper, we propose a lightweight convolution attention model named as CTFALite, which learns channel-specific temporal attention and frequency attention by leveraging both of the global context information and the local cross-channel dependencies. Experiment results demonstrate the effectiveness of CTFALite for improving performance. The further analysis about computational resource consumption shows that CTFALite achieves a better trade-off between performance and computational complexity, compared to other competing lightweight convolution attention mechanisms.

Index Terms: convolution attention mechanism, lightweight attention model, low resource consumption, speaker recognition

1. Introduction

Speaker recognition aims to determine the identities of speakers based on their utterances, which involves two sub-tasks speaker verification and speaker identification [1]. The advances in speaker recognition are mainly attributed to exploring how to extract more discriminative speaker representations (i-vector [2], d-vector [3], x-vector [4], etc.) and how to construct more powerful similarity estimators (Probabilistic Linear Discriminant Analysis [5][6], Neural PLDA [7], etc.).

Modern speaker recognition systems obtain significant performance improvement with the help of deep learning technologies. Deep neural networks are used as backbones to extract speaker representation known as speaker embedding from raw utterances. Time delay neural network (TDNN) [8] and residual neural network (ResNet) [9] are two of the most popular network structures of speaker embedding extractors [4][10]. The powerful representation learning capability of deep neural networks is crucial to capturing discriminative speaker characteristics from speech signals, which is largely determined by network architectures. Recently, several elaborate structures are proposed to enhance this capability, such as ECAPA-TDNN [11] and SE-SCNet [12], which integrate advanced convolution blocks (Res2Block [13], SC-Block [14], etc.) and multiple-scale feature aggregation methods into TDNN or ResNet. In addition to directly reforming the neural network architectures, some studies incorporate attention mechanisms into mature deep neural networks to improve their representation learning capability. Attention models are used to selectively emphasize features with more speaker characteristics and to capture latent dependencies among features.



(a) SE

(b) ECA

(c) tf-CBAM

(d) CA

Figure 1: *Convolution attention mechanisms: (a) SE, (b) ECA, (c) tf-CBAM, and (d) CA. GAP and GMP refer to global average pooling and global max pooling, respectively. AP and MP denote direction-specific average pooling and max pooling, respectively. BN refers to batch normalization. r is a dimensionality-reduction ratio w.r.t the number of channels. FC is a fully connected layer. Conv is a convolution layer. c, f and t refer to the channel, frequency and time directions, respectively.*

Most previous studies have investigated attention mechanism based pooling layers, which aggregate a sequence of frame-level feature vectors into an utterance-level speaker embedding after assigning attention weights to every frames, such as self-attention pooling [15], multi-head attention pooling [16]

[*]Corresponding author

10.21437/Interspeech.2022-10288

and multi-resolution multi-head attention pooling [17]. Different from them, we focus on convolution attention mechanisms, which enhance the convolution blocks by generating attention weights for speaker feature tensors with the shape of $(channel, frequency, time)$. Though the computation-intensive Non-local attention [18] can thoroughly model the long-range dependencies among all of the pixel-level features in a feature tensor, light-weight convolution attention models are more suitable to low-resource edge devices (smartphone, iPad, etc.), such as Squeeze-and-Excitation Network (SE) [19], Efficient Channel Attention (ECA) [20] and Convolutional Block Attention Module (CBAM) [21]. While SE and ECA only learn channel attention for feature tensors, CBAM additionally learns spatial attention. When used for speaker embedding extractors, CBAM fuses the frequency direction and the time direction to calculate spatial attention weights, which ignores the difference between these two directions. To address the issue, Yadav *et.al* [22] proposed tf-CBAM, which improves CBAM by separately learning frequency attention and time attention. However, tf-CBAM forces each channel of a feature tensor to share the same temporal and frequency attention weights. We believe this can not exactly capture the inter-dependencies among features in a 3D feature tensor. In the computer vision field, recently, coordinate attention (CA) [23] is proposed to separately encode information from two different spatial directions (height and width) into channels and to learn direction-aware and position-sensitive channel attention. We employ CA to learn channel-specific attention for speaker embedding extractors. However, the authors of ECA [20] found that it destroys the direct correspondence between channels and attention weights to control model complexity by decreasing the number of channels. CA also uses the trick to reduce its computational complexity, which imports a latent defect. Besides, the computational burden of CA is much higher than SE, ECA and tf-CBAM.

In this paper, we propose a light-weight attention mechanism named as *CTFALite*, which integrates global context information and local cross-channel interaction to learn channel-specific temporal and frequency attention. Our contributions are summarized as follows: (1) We introduce the CA mechanism into speaker embedding extractors and demonstrate its effectiveness for speaker recognition. (2) We propose a low-complexity CTFALite mechanism to enhance the representation learning capability of speaker embedding extractors. CTFALite reaches a better trade-off between recognition performance and computational resource consumption, compared to SE, ECA, tf-CBAM and CA. (3) We conduct comprehensive experiments to evaluate light-weight convolution attention mechanisms. Code is available at https://github.com/star9012/CTFALite.

2. Convolution Attention Mechanisms

We briefly review four light-weight convolution attention mechanisms, including SE, ECA, tf-CBAM and CA. These convolution attention models take as input a 3D feature tensor $X \in \mathbb{R}^{N_c \times N_f \times N_t}$, where N_c, N_f and N_t denote the channel dimension, the frequency dimension and the time dimension, respectively. Let $\odot$ denote broadcast point-wise multiplication.

2.1. SE

SE [19] is proposed to explicitly model the inter-dependencies among the channels of X, which consists of a squeeze operation and an excitation operation. As shown in Figure 1(a), the squeeze operation adopts global average pooling to aggregate X across its spatial directions ($N_f \times N_t$), so that the global spatial information of channel-wise feature responses is encoded into an embedding with the length of N_c. The excitation operation transforms the embedding to a collection of channel attention weights with two non-linear fully connected layers. During the transformation, the length of the embedding is decreased to $\frac{N_c}{r}(r > 1)$ by one layer and resumed to N_c by the other layer, which effectively reduces the model complexity. Let $W_c \in \mathbb{R}^{N_c \times 1 \times 1}$ refer to the channel attention weights given by SE, the enhanced feature tensor is generated by $X \odot W_c$.

2.2. ECA

ECA [20] is an efficient channel attention mechanism based on local cross-channel interaction. The authors of ECA dissected SE and found the side effect of dimensionality reduction on channel attention modelling. To address this issue, ECA employs a convolution operator of size k and stride 1 to calculate attention weights, so that the attention weight of the channel c_i depends on k neighbors of c_i, as shown in Fig 1(b).

2.3. tf-CBAM

As shown in Fig.1(c), tf-CBAM [22] generates the channel attention weights $W_c \in \mathbb{R}^{N_c \times 1 \times 1}$ by the same process as SE except that tf-CBAM simultaneously uses global average pooling and global max pooling to aggregate spatial features. The channel-enhanced feature tensor $X_c = X \odot W_c$ is input to subsequent operators to separately model frequency attention and time attention. When calculating the attention weights w.r.t one specific spatial direction (time or frequency), tf-CBAM squeezes X_c along the other spatial direction and the channel direction by pooling operations. Finally, convolution operators are used to generate the frequency attention weights $W_f \in \mathbb{R}^{1 \times N_f \times 1}$ and the temporal attention weights $W_t \in \mathbb{R}^{1 \times 1 \times N_t}$. The tf-CBAM enhanced feature tensor $X_{\text{tf-CBAM}}$ is as follows:

$$X_{\text{tf-CBAM}} = \frac{1}{2}(X_c \odot W_f + X_c \odot W_t). \quad (1)$$

2.4. CA

Different from tf-CBAM modelling channel-independent frequency attention and time attention, CA can separately embed frequency and temporal information into channels. Fig.1(d) shows the details of CA attention. CA directly squeezes X along one specific spatial direction by average pooling, so that the other spatial direction and the channel direction are kept. Therefore, the generated two feature tensors capture direction-specific information. For computational efficiency, these two feature tensors are concatenated along the spatial dimension and sent to a shared fully connected layer. Then, the output is split into two feature tensors along the spatial dimension. Finally, two independent fully connected layers are used to generate the frequency attention weights $W_f \in \mathbb{R}^{N_c \times N_f \times 1}$ and the temporal attention weights $W_t \in \mathbb{R}^{N_c \times 1 \times N_t}$, respectively. The CA-enhanced feature tensor X_{CA} is generated as follows:

$$X_{\text{CA}} = X \odot W_f \odot W_t. \quad (2)$$

3. Lightweight Channel-specific Temporal and Frequency Attention Mechanism

Though CA reduces the number of channels by a bottleneck layer, its computational complexity is still relatively higher than

Table 1: *Equal error rates (%) of different attention mechanisms on the official trials of VoxCeleb1 (Vox1) and CnCeleb (Cn). Self-AP: Self-Attentive Pooling. TAP: Temporal Average Pooling. SAP: Statistical Average Pooling. MHA: Multi-Head Attention. MRMHA: Multiple-Resolution Multiple-Head Attention. ASP: Attentive Statistics Pooling. The hyper-parameter r is set to* 4

	Backbone Model	Convolutional Attention	Aggregation	Dims	Vox1-EER	Cn-EER
Cai *et al.* [24]	ResNet-34	-	Self-AP	128	4.40	-
Hajibabei *et al.* [10]	ResNet-29	-	TAP	128	4.30	-
India *et al.* [16]	CNN	-	MHA	-	4.00	-
Wang *et al.* [17]	ResNet-34	-	MRMHA	512	3.97	-
Okabe *et al.* [25]	TDNN	-	ASP	1500	3.85	-
Our Experiments	PA-ResNet50	-	TAP	256	4.008	15.306
	PA-ResNet50	SE	TAP	256	3.860	15.112
	PA-ResNet50	ECA	TAP	256	3.791	14.982
	PA-ResNet50	tf-CBAM	TAP	256	3.685	14.559
	PA-ResNet50	CA	TAP	256	3.351	13.821
	PA-ResNet50	CTFALite	TAP	256	3.515	13.995
Our Experiments	PA-ResNet50	-	SAP	256	4.173	16.266
	PA-ResNet50	SE	SAP	256	4.247	16.091
	PA-ResNet50	ECA	SAP	256	4.130	16.639
	PA-ResNet50	tf-CBAM	SAP	256	3.887	15.587
	PA-ResNet50	CA	SAP	256	3.759	14.678
	PA-ResNet50	CTFALite	SAP	256	3.696	15.263
Our Experiments	PA-ResNet50	-	MHA	256	3.929	15.082
	PA-ResNet50	SE	MHA	256	4.067	15.538
	PA-ResNet50	ECA	MHA	256	4.056	15.567
	PA-ResNet50	tf-CBAM	MHA	256	3.812	15.781
	PA-ResNet50	CA	MHA	256	3.383	14.654
	PA-ResNet50	CTFALite	MHA	256	3.436	14.278
Our Experiments	PA-ResNet50	-	MRMHA	256	3.860	15.140
	PA-ResNet50	SE	MRMHA	256	3.818	15.274
	PA-ResNet50	ECA	MRMHA	256	3.977	15.500
	PA-ResNet50	tf-CBAM	MRMHA	256	3.606	15.505
	PA-ResNet50	CA	MRMHA	256	3.324	13.722
	PA-ResNet50	CTFALite	MRMHA	256	3.404	13.213

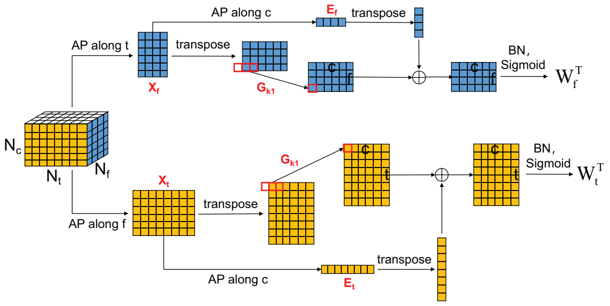


Figure 2: *CTFALite Attention. AP: average pooling. t: temporal direction. f: frequency direction. c: channel direction. BN: batch normalization. ⊕: broadcast point-wise addition.*

SE, ECA and tf-CBAM. Additionally, the dimensionality reduction trick destroys the direct connection between attention weights and channels. CTFALite adopts small-size convolution to control the model complexity without decreasing the channel dimension of $X \in \mathbb{R}^{N_c \times N_f \times N_t}$, as ECA [20]. However, the convolution operator focuses on local cross-channel interaction, which is not good at capturing long-range dependencies. We create global context embeddings to compensate for the lack of global information about all the channels of X.

Let $\odot$ and $\oplus$ refer to broadcast point-wise multiplication and addition, respectively. The proposed CTFALite attention model is shown in Fig. 2. CTFALite firstly uses average pooling to compress X into two feature maps $X_f \in \mathbb{R}^{N_c \times N_f}$ and $X_t \in \mathbb{R}^{N_c \times N_t}$ w.r.t two spatial directions (frequency and time). For each of X_f and X_t, CTFALite aggregates all of its channels to form a global context embedding with an average pooling operator along the channel direction, which generates $E_f \in \mathbb{R}^{1 \times N_f}$ and $E_t \in \mathbb{R}^{1 \times N_t}$. CTFALite uses $1 \times k_1$ convolution G_{k1} to capture the cross-channel local dependencies. Global context information and local dependencies are fused to generate the attention weights. The channel-specific frequency and temporal attention weights $W_f^T \in \mathbb{R}^{N_f \times N_c}$ and $W_t^T \in \mathbb{R}^{N_t \times N_c}$ are respectively generated by Eq.3 and Eq.4, where σ is a sigmoid function, BN denotes batch normalization and T refers to matrix transpose. As ECA [20], the kernel size k_1 is adaptively determined by N_c. We set the value of k_1 based on Eq.5, where $||_{odd}$ ensures that k_1 is an odd number.

$$W_f^T = \sigma(BN(G_{k1}(X_f^T) \oplus E_f^T)) \tag{3}$$

$$W_t^T = \sigma(BN(G_{k1}(X_t^T) \oplus E_t^T)) \tag{4}$$

$$k_1 = |\log_2(N_c)|_{odd} \tag{5}$$

Let reshape W_f and W_t to $W_f \in \mathbb{R}^{N_c \times N_f \times 1}$ and $W_t \in \mathbb{R}^{N_c \times 1 \times N_t}$. The enhanced feature tensor X_{CTFALite} is

$$X_{\text{CTFALite}} = X \odot W_f \odot W_t. \tag{6}$$

4. Experiments

4.1. Experimental Setup

Dataset: We train speaker embedding extractors with the training set of VoxCeleb1 [26] and evaluate them with the official trials provided by VoxCeleb1 and CnCeleb [27].

Acoustic Feature: During training, we randomly extract 2.5 s segments from training utterances and perform the 320-point Fast Fourier Transform (FFT) algorithm [28] on them to create spectrograms. When performing FFT, we use a sliding window of 20 ms length and 10 ms stride to divide each utterance segment into multiple frames. The mean-and-variance normalization is applied to the generated spectrograms.

DNN Structure: We use a pre-activation residual neural network of 50 layers (PA-ResNet50) [29] as the speaker embedding extractor in our experiments, as in [22]. The size of the speaker embedding is set to 256. ReLu activation function [30] and batch normalization [31] are used in PA-ResNet50.

Loss Function: We use the additive angular margin loss function (AAM-Softmax) [32] to train the speaker embedding extractor. And the hyper-parameter m w.r.t the angular margin penalty is dynamically updated as follows:

$$m = m_{max} \times (\frac{2}{1 + \exp(-10 \times p)} - 1), \tag{7}$$

where p increases from 0 to 1 during the whole training process and m_{max} is set to 0.15 in our experiments.

Training strategy: The experiments are conducted based on PyTorch [33]. We use the Adam optimizer [34] with an initial learning rate of $1e$-3 to train models. The learning rate is divided by 10 every 40 epochs until the model convergence. The weight decay is set to $1e$-4 and the batch size is 64.

4.2. Results

Comparison with Convolution Attention Mechanisms: We compare CTFALite with four competing low-complexity convolution attention mechanisms, including SE, ECA, tf-CBAM and CA. We also combine these convolution attention models with different back-end pooling layers to observe performance changes. For a fair comparison, all of the methods adopt the same extractor structure and are trained with identical optimisation schemes. We use cosine similarity to measure the similarity between a pair of speaker embeddings. Equal error rate (EER) is used as the performance metric of different methods.

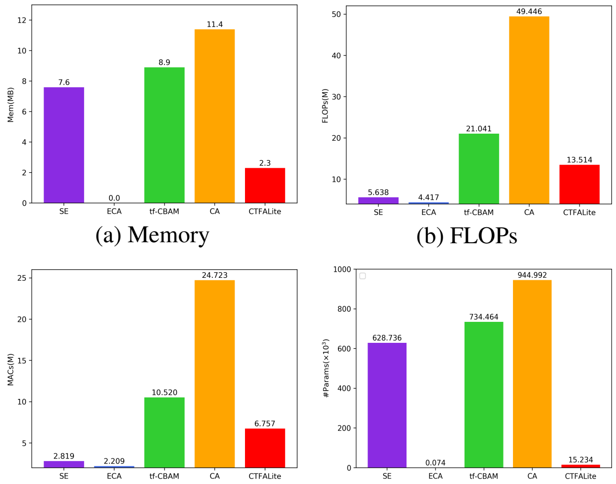
Table. 1 lists the evaluation results. We observe that all of these tested convolution attention mechanisms conduce to performance improvement. Compared to SE and ECA which only focus on channel attention, tf-CBAM and CA achieve better results because they additionally learn temporal attention and frequency attention. CA outperforms tf-CBAM, which demonstrates the effectiveness of learning channel-specific spatial attention weights. The proposed CTFALite attention mechanism achieves competing performance, compared to CA.

Ablation Study: We conduct ablation experiments for CTFALite as follows: (1) *V1*: the global context embeddings E_f and E_t are not used. (2) *V2*: the temporal attention W_t is not used. (3) *V3*: the frequency attention W_f is not used. We use a statistical pooling layer to aggregate frame-level features. The results are reported in Table.2. These variants of CTFALite obtain worse results, which demonstrates that each component of CTFALite contributes to performance improvement.

Computational Resource Consumption: We evaluate the computational resource consumption of convolution attention models. The evaluation metrics include memory usage (Mem) for storing models, floating point operations (FLOPs), multiply–accumulate operations (MACs), and the total number of model parameters (#Params). Fig. 3 shows the total resource consumption of convolution attention layers when a feature tensor with the size of $(1, 161, 248)$ is taken as input by a speaker embedding extractor. As can be seen, SE and ECA have the smallest computational burden, but they also achieve the smallest performance gain in our experiments. CA obtains the best experimental results, but its computational complexity is significantly higher than other attention models. The proposed CTFALite model presents a better trade off between recognition performance and computational resource consumption.

Table 2: *Ablation experiments for CTFALite. EER (%) is measured on the test set of VoxCeleb1.*

	CTFALite	V1	v2	V3
EER	3.696	3.888	4.093	4.099



(a) Memory (b) FLOPs

(c) MACs (d) #Params

Figure 3: *Computational resource consumption of convolution attention mechanisms. Here, the zero value of memory usage means the memory consumption is quite slight.*

5. Conclusion

In this paper, we propose a lightweight convolution attention mechanism CTFALite to enhance the speaker embedding extractors for speaker recognition on edge devices. CTFALite separately learns channel-specific temporal and frequency attention by capturing both local and global cross-channel dependences. Experimental results demonstrate the effectiveness of CTFALite for improving performance and also show the low-resource consumption characteristic of CTFALite.

6. Acknowledgements

This work is partially supported by a grant from the National Natural Science Foundation of China(No.62032017), the Key Industrial Innovation Chain Project in Industrial Domain of Shaanxi Province (No.2021ZDLGY03-09, No.2021ZDLGY07-02, No.2021ZDLGY07-03) and The Youth Innovation Team of Shaanxi Universities.

7. References

[1] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[2] N. Dehak, P. 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.

[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] S. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007*, 2007.

[6] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "Plda for speaker verification with utterances of arbitrary duration," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7649–7653.

[7] S. Ramoji, P. Krishnan, and S. Ganapathy, "Nplda: A deep neural plda model for speaker verification," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 202–209.

[8] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth annual conference of the international speech communication association*, 2015.

[9] 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.

[10] M. Hajibabaei and D. Dai, "Unified hypersphere embedding for speaker recognition," *arXiv preprint arXiv:1807.08312*, 2018.

[11] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *Proc. Interspeech 2020*, pp. 3830–3834, 2020.

[12] Y.-J. Zhang, Y.-W. Wang, C.-P. Chen, C.-L. Lu, and B.-C. Chan, "Improving time delay neural network based speaker recognition with convolutional block and feature aggregation methods," *Proc. Interspeech 2021*, pp. 76–80, 2021.

[13] 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.

[14] J.-J. Liu, Q. Hou, M.-M. Cheng, C. Wang, and J. Feng, "Improving convolutional networks with self-calibrated convolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 096–10 105.

[15] F. R. rahman Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5359–5363.

[16] M. À. India Massana, P. Safari, and F. J. Hernando Pericás, "Self multi-head attention for speaker recognition," in *Interspeech 2019: the 20th Annual Conference of the International Speech Communication Association: 15-19 September 2019: Graz, Austria*. International Speech Communication Association (ISCA), 2019, pp. 4305–4309.

[17] 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.

[18] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[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] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 531–11 539.

[21] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[22] S. Yadav and A. Rai, "Frequency and temporal convolutional attention for text-independent speaker recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6794–6798.

[23] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 713–13 722.

[24] W. Cai, J. Chen, and L. Ming, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Odyssey 2018*, 2018.

[25] O. Koji, K. Takafumi, and S. Koichi, "Attentive statistics pooling for deep speaker embedding," *Proc. Interspeech 2018*, pp. 2252–2256, 2018.

[26] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

[27] 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.

[28] J. W. Cooley, P. A. Lewis, and P. D. Welch, "The fast fourier transform and its applications," *IEEE Transactions on Education*, vol. 12, no. 1, pp. 27–34, 1969.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[30] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[31] 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.

[32] 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.

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.