



Improved CNN-Transformer Using Broadcasted Residual Learning for Text-Independent Speaker Verification

Jeong-Hwan Choi, Joon-Young Yang, Ye-Rin Jeoung, and Joon-Hyuk Chang

Department of Electronic Engineering, Hanyang University, Seoul, Republic of Korea

{brent1104, dreadbird, jyr0328, jchang}@hanyang.ac.kr

Abstract

This study proposes a novel speaker embedding extractor architecture that effectively combines convolutional neural networks (CNNs) and Transformers. Based on the recently proposed *CNNs-meet-vision-Transformers* (CMT) architecture, we propose two strategies for efficient speaker embedding extraction modeling. First, we apply broadcast residual learning techniques to the building blocks of the CMT, allowing us to extract frequency-aware temporal features shared across frequency dimensions with a reduced set of parameters. Second, frequency-statistics-dependent attentive statistics pooling is proposed to aggregate attentive temporal statistics acquired from the means and standard deviations of input feature maps weighted along the frequency axis using an attention mechanism. The experimental results on the VoxCeleb-1 dataset show that the proposed model outperforms several CNN- and Transformer-based models with a similar number of model parameters. Moreover, the effectiveness of the proposed modifications to the CMT architecture is validated through ablation studies.

Index Terms: Text-independent speaker verification, Transformer, hybrid deep neural network, attentive statistics pooling

1. Introduction

Modern text-independent speaker verification (TI-SV) systems typically employ a speaker embedding extractor that can encode speaker-discriminative features from a variable-duration utterance. With the emergence of deep learning, various speaker embedding extractors have been proposed, including time-delay neural networks (TDNNs), represented by X-vectors [1], and ResNet-based models [2]. To further enhance the feature extraction ability of convolutional neural networks (CNNs), densely connected convolution [3] layers and the Res2Net [4] architecture were adopted in [5] and [6], respectively, to improve the TDNN model. Recently, to enhance its capability in extracting global features, attempts have been made to integrate the Transformer [7] architecture into a speaker embedding extractor [8, 9]. In addition to the feature extractor architecture, the temporal pooling mechanism, which transforms a variable-duration feature sequence into a fixed-dimensional speaker embedding, has evolved from simple statistics pooling [1] to attentive [10, 11] and channel-dependent [6] versions.

In a computer vision field, several studies have combined CNNs and Transformers to process 2D images [12–14]. In particular, *CNNs-meet-vision-Transformers* (CMT) [14] is a hybrid architecture designed to capture local features using CNNs and long-range dependencies using Transformer. The CMT comprises a stack of several network blocks; each of which contains a local perception unit (LPU), a lightweight multi-head self-attention (LMHSA) module, and an inverted residual feed-forward network (IRFFN) module [14]. These modules achieve more efficient computations than the Transformers.

In this study, we introduce a CMT-based speaker embedding extractor with an improved structure. Inspired by broadcasted residual learning (BRL) [15], we first propose broadcasted (BC)-CMT, which substitutes the 2D convolution in CMT modules with two separated convolution layers. Each of these layers operates frequency- or temporal-wise to extract dimension-wise local features with fewer model parameters. We also propose frequency-statistics-dependent attentive statistics pooling (FS-ASP) to leverage statistical information in the frequency dimension. The FS-ASP utilizes attention mechanisms in a cascade manner along the frequency and time axes of an input feature map. The proposed methods are evaluated and compared with other models on the VoxCeleb TI-SV benchmark [16, 17].

2. CMT architecture

2.1. CMT block

As described briefly in Section 1, a CMT block is modularized with a cascade of LPUs, LMHSAs, and IRFFNs to alternately capture local and global structural information. The LPU comprises a depthwise convolution with a residual connection and extracts local information from an input feature map.

$$\text{LPU}(\mathbf{X}) = \text{DWConv}(\mathbf{X}) + \mathbf{X}, \quad (1)$$

where $\mathbf{X} \in \mathbb{R}^{S \times C}$ denotes an input feature map of size S and number of channels C . Also, $\text{DWConv}(\cdot)$ denotes the depthwise convolution.

The LMHSA is a lightweight version of the MHSA [7], which is designed to perform the MHSA with fewer computations. In [7], an input feature map is linearly transformed into a query $\mathbf{Q} \in \mathbb{R}^{S \times C_k}$, key $\mathbf{K} \in \mathbb{R}^{S \times C_k}$, and value $\mathbf{V} \in \mathbb{R}^{S \times C_v}$. In comparison, the LMHSA substitutes linear transformations with $k \times k$ depthwise convolutions with a stride size of k [14]. It decreases the spatial size of the feature map, and thus, reduces the number of computations required for the MHSA.

$$\text{LMHSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{C_k}} + \mathbf{B}\right)\mathbf{V}, \quad (2)$$

where $\mathbf{B} \in \mathbb{R}^{S \times \frac{S}{k^2}}$ is a learnable parameter, referred to as the relative position bias, and interacts with \mathbf{Q} and \mathbf{K} [14].

The IRFFN substitutes two dense layers of the feed-forward network module of the original Transformer [7] with three convolutions with different roles. The first pointwise convolution expands the channel size of a feature map by a factor of R , which is followed by depthwise convolution with kernel sizes of 3×3 to capture local information in the high-dimensional feature space [14]; this only adds negligible amounts of computations. Subsequently, a second pointwise convolution is applied to reduce the channel size by $\frac{1}{R}$ such that

$$\text{IRFFN}(\mathbf{X}) = \text{PWConv}(f(\text{GELU}(\text{PWConv}(\mathbf{X})))), \quad (3)$$

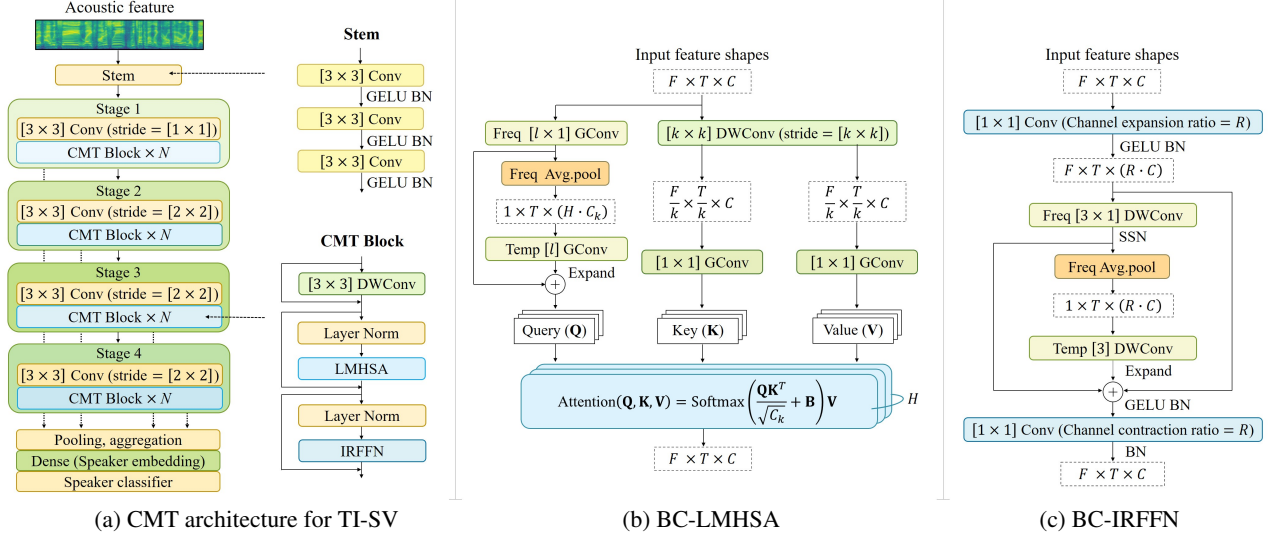


Figure 1: Proposed speaker embedding extractor architecture based on CMT with BRL.

$$f(\mathbf{X}) = \text{GELU}(\text{DWConv}(\mathbf{X}) + \mathbf{X}), \quad (4)$$

where $\text{PWConv}(\cdot)$ and $\text{GELU}(\cdot)$ denote the pointwise convolution and GELU activation [18], respectively. Note that batch normalization [19] is applied after every activation and final convolution.

2.2. CMT-based speaker embedding extractor

Fig. 1(a) shows the proposed CMT-based speaker embedding extractor comprising a stem, stages, a pooling layer, and a speaker classifier. The stem is constructed with three convolutions to extract fine-grained features, and the stages are designed with N CMT blocks with the same topology. The first layer of each stage scales down the input feature map using 3×3 depthwise convolutions to extract multi-scale features, which is followed by a stack of CMT blocks. To handle variable-duration sequences of input features, the relative position bias, $\mathbf{B} \in \mathbb{R}^{F \times \frac{F}{k^2} \times T}$, in Eq. (2) is broadcast along the time frame dimension T of an input feature map. The number of CMT blocks and the downsampling of the feature map sizes are set with reference to the ResNet-based speaker embedding extractor [2], to perform a speaker classification task. Section 4 details the proposed architecture. The output feature map of each stage (Fig. 1(a)) is used to compute speaker embedding via a pooling layer, followed by a dense layer. The output layer is another dense layer that predicts the speaker posteriors of training speakers.

3. Proposed BC-CMT block and FS-ASP

3.1. CMT block with BRL

This section discusses BRL [15] and the structure of the proposed BC-CMT block, including BC-LMHSA and BC-IRFFN. In general, deep residual learning can be expressed as the sum of a shortcut and the output of a convolution, which are usually in the same dimension. In comparison, BRL was proposed to extract frequency and temporal features separately with different shapes of convolutions that operate in different dimensions [15]. Specifically, BRL decomposes a 2D frequency \times temporal convolution with kernel sizes of $l \times l$ into a 1D temporal convolution

and 2D frequency-wise convolution with kernel sizes of l and $l \times 1$, respectively [15].

$$\text{BRL}(\mathbf{X}) = \text{BC}(\text{T}(\text{AVGpool}(\text{F}(\mathbf{X})))) + \mathbf{X} + (\text{F}(\mathbf{X}))^{(\text{optional})}, \quad (5)$$

where $\text{T}(\cdot)$, $\text{F}(\cdot)$, $\text{AVGpool}(\cdot)$, and $\text{BC}(\cdot)$ denote temporal, frequency, average, and BC operations, respectively. Because the temporal convolution proceeds after the cascade of a frequency-wise convolution and an average pooling (across the frequency axis), it produces frequency-aware temporal features. The BC operation broadcasts a 1D temporal feature map along the frequency dimension prior to the shortcut connection. Batch normalization with nonlinear activation and subspectral normalization [20] are applied after the temporal and frequency-wise convolution, respectively. In addition, an auxiliary residual connection with a frequency-wise convolution can be used to achieve frequency awareness [15]. To summarize, BRL squeezes the frequency information of a feature map, calibrates it with a temporal convolution, and broadcasts it along the frequency axis with a residual connection. Note that BRL operates with fewer parameters than deep residual learning with typical 2D convolution layers.

The BC-LMHSA is designed to exploit both BRL and LMHSA, thus alternately capturing frequency-aware temporal features with long-range dependencies. Specifically, Fig. 1(b) shows that the linear operation for \mathbf{Q} in [14] is substituted with convolution operations through BRL. Additionally, a frequency-wise and temporal group convolution with kernel sizes of $l \times 1$ and l , respectively, are used, where l is set to be similar to the reduction rate k in LMHSA. Moreover, because the size of an input feature map $\mathbf{X} \in \mathbb{R}^{F \times T \times C}$ is typically different from that of $\mathbf{Q} \in \mathbb{R}^{F \times T \times (H \times C_k)}$, only the auxiliary residual connection is used.

$$\mathbf{Q}(\mathbf{X}) = \text{BC}(\text{T}_g(\text{AVGpool}(\text{F}_g(\mathbf{X})))) + \text{F}_g(\mathbf{X}), \quad (6)$$

where F_g and T_g denote the frequency-wise and temporal group convolutions, respectively. The linear operations for \mathbf{K} and \mathbf{V} in [14] are replaced with 1×1 group convolutions to make the LMHSA lighter than original CMT. Note that nonlinearity is not applied for computing \mathbf{Q} , \mathbf{K} , and \mathbf{V} . The self-attention in the BC-LMHSA operates as described in Eq. (2).

Table 1: Architectures of proposed BC-CMT-based speaker embedding extractors. Options for BC-CMT blocks are shown in brackets with numbers of blocks stacked, N . H , k , and l denote number of attention heads, reduction rate, and kernel size of group convolution in BC-LMHSA, respectively. T denotes duration of input feature map.

Output size	Name	BC-CMT-Tiny	BC-CMT-Small	BC-CMT-Base
$80 \times T$	Stem	$[3 \times 3, 8] \times 3$	$[3 \times 3, 16] \times 3$	$[3 \times 3, 32] \times 3$
Stage 1 $80 \times T$	DW Conv.	$3 \times 3, 8, \text{stride } 1$	$3 \times 3, 16, \text{stride } 1$	$3 \times 3, 32, \text{stride } 1$
	$\begin{bmatrix} \text{LPU} \\ \text{BC-LMHSA} \end{bmatrix} \times N$	$\begin{bmatrix} 3 \times 3, 8 \\ H=1, k=8, l=9 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 16 \\ H=1, k=8, l=9 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 32 \\ H=1, k=8, l=9 \end{bmatrix} \times 3$
Stage 2 $40 \times \frac{T}{2}$	DW Conv.	$3 \times 3, 16, \text{stride } 2$	$3 \times 3, 32, \text{stride } 2$	$3 \times 3, 64, \text{stride } 2$
	$\begin{bmatrix} \text{LPU} \\ \text{BC-LMHSA} \end{bmatrix} \times N$	$\begin{bmatrix} 3 \times 3, 16 \\ H=2, k=4, l=5 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 32 \\ H=2, k=4, l=5 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ H=2, k=4, l=5 \end{bmatrix} \times 3$
Stage 3 $20 \times \frac{T}{4}$	DW Conv.	$3 \times 3, 32, \text{stride } 2$	$3 \times 3, 64, \text{stride } 2$	$3 \times 3, 128, \text{stride } 2$
	$\begin{bmatrix} \text{LPU} \\ \text{BC-LMHSA} \end{bmatrix} \times N$	$\begin{bmatrix} 3 \times 3, 32 \\ H=4, k=2, l=3 \end{bmatrix} \times 6$	$\begin{bmatrix} 3 \times 3, 64 \\ H=4, k=2, l=3 \end{bmatrix} \times 10$	$\begin{bmatrix} 3 \times 3, 128 \\ H=4, k=2, l=3 \end{bmatrix} \times 16$
Stage 4 $10 \times \frac{T}{8}$	DW Conv.	$3 \times 3, 64, \text{stride } 2$	$3 \times 3, 128, \text{stride } 2$	$3 \times 3, 256, \text{stride } 2$
	$\begin{bmatrix} \text{LPU} \\ \text{BC-LMHSA} \end{bmatrix} \times N$	$\begin{bmatrix} 3 \times 3, 64 \\ H=8, k=1, l=1 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ H=8, k=1, l=1 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ H=8, k=1, l=1 \end{bmatrix} \times 3$
1×1	Embedding	480×128	960×512	1920×512

Fig. 1(c) demonstrates that the BC-IRFFN applies BRL to efficiently reduce the number of parameters of a depthwise convolution, and operates in the expanded channel dimension $R \cdot C$. The residual connection of the original IRFFN in Eq. (3) is substituted with two separate convolutions and two shortcuts.

$$f(\mathbf{X}) = \text{GELU}(\text{BC}(\text{T}_{\text{dw}}(\text{AVG}_{\text{pool}}(\text{SSN}(\text{F}_{\text{dw}}(\mathbf{X})))))) + \text{SSN}(\text{F}_{\text{dw}}(\mathbf{X})) + \mathbf{X}, \quad (7)$$

where $\text{F}_{\text{dw}}(\cdot)$ and $\text{T}_{\text{dw}}(\cdot)$ denote the frequency-wise and temporal depthwise convolutions with kernel sizes of 3×1 and 3, respectively. Also, $\text{SSN}(\cdot)$ denotes subspectral normalization [20].

3.2. FS-ASP with multi-stage pooling aggregation (MSPA)

The BC-CMT architecture broadcasts the frequency-aware temporal feature across the frequency axis on a block-by-block basis. To effectively capture the speaker information available in the frequency dimension, we propose the FS-ASP, which performs channel-dependent ASP [6] across both the frequency and temporal dimensions. Specifically, an attention mechanism is applied across the frequency axis at every time step to compute the frequency-wise attention scores as follows:

$$\alpha_{f,t,c} = \frac{\exp(\text{MLP}_c(\mathbf{X}_{f,t}))}{\sum_c^F \exp(\text{MLP}_c(\mathbf{X}_{\zeta,t}))}, \quad (8)$$

where $\mathbf{X}_{f,t} \in \mathbb{R}^C$ denotes an input feature map sliced at the f -th frequency element and t -th frame, and C denotes the number of channels. In addition, $\text{MLP}_c(\cdot)$ denotes a multilayer perceptron, comprising two dense layers [6] and a nonlinearity between them. Score $\alpha_{f,t,c}$ represents the importance of each frequency element given channel c at the time step t . The channel-dependent weighted mean and standard deviation are then estimated for each time step t as follows:

$$\bar{\mu}_{t,c} = \sum_f^F \alpha_{f,t,c} \mathbf{X}_{f,t,c}, \quad \bar{\sigma}_{t,c} = \sqrt{\sum_f^F \alpha_{f,t,c} \mathbf{X}_{f,t,c}^2 - \bar{\mu}_{t,c}^2}. \quad (9)$$

Subsequently, for each of these statistics, temporal attention scores are computed across the temporal axis, which are em-

ployed to produce temporally weighted mean and standard deviations. Consequently, four vectors in \mathbb{R}^C are obtained and concatenated into a single vector.

We employ an MSPA [21], which concatenates the outputs of stage-dependent pooling operations, to extract statistical features at different resolutions and depths.

4. Experimental setup

4.1. Implementation

The input acoustic features were 2.5 s random crops of 80-dimensional log mel-filterbank energies, which were extracted with a 25 ms window and 10 ms shift; voice activity detection was not applied. The details of the architecture are listed in Table 1; three versions of the BC-CMT with different sizes were proposed. The model sizes of the BC-CMT-Base and BC-CMT-Small were determined by modifying the number of stacked blocks N and convolution channel dimensions. Both models had 512D embeddings. In comparison, the BC-CMT-Tiny was designed as a lightweight model with an embedding size of 128. For the BC-LMHSA, the number of convolution groups and C_k were set as depthwise convolution for all models. The expansion ratio R of the BC-IRFFN was set to 4.0 for the BC-CMT-Base and 3.6 for the BC-CMT-Tiny and BC-CMT-Small. AAM-softmax [22] with a margin of 0.22 and scale of 35 was adopted to train all systems, and the Adam optimizer [23] with a mini-batch size of 60 was used. The learning rate was initially set to 0.001, and decayed by half when the validation loss plateaued three times. To prevent overfitting, ℓ_2 -regularization was applied to all the weights with a scaling factor of $5e-4$.

4.2. Dataset and evaluation protocol

The proposed models were trained and evaluated using the VoxCeleb corpus. The development part of the VoxCeleb-2 dataset with 5,994 speakers was used for training. Two additional samples were generated for each utterance via speed perturbation with factors of 0.9 and 1.1 for data augmentation purposes. From 1,500 speakers, 6,000 utterances were randomly selected from the training set for the validation process. The models were evaluated using the trial sets of the cleaned VoxCeleb-1 dataset, and speaker-wise adaptive score normalization [6, 27]

Table 2: TI-SV results of proposed BC-CMT models on the VoxCeleb-1 original, extended, and hard test sets

Models	# Params	VoxCeleb-1 O		VoxCeleb-1 E		VoxCeleb-1 H	
		EER(%)	minDCF _{0.05}	EER(%)	minDCF _{0.05}	EER(%)	minDCF _{0.05}
BC-CMT-Tiny	273.6K	2.70	0.175	2.71	0.162	4.16	0.228
BC-CMT-Small	1.4M	1.05	0.061	1.19	0.074	2.03	0.115
BC-CMT-Base	6.3M	0.86	0.049	1.10	0.067	1.85	0.106

Table 3: EER(%) and minDCF_{0.01} comparison of various models with different sizes which were trained using VoxCeleb corpus (* our implementation).

Models	# Params	VoxCeleb-1 O	
		EER	minDCF
Julien <i>et al.</i> [24]	237.5K	3.31	-
ECAPA-TDNNLite [25]	318K	3.07	0.296
SAEP [8]	1.2M	5.44	-
Fast-ResNet-34 [26] *	1.4M	2.17	0.195
D-TDNN-SS [5]	3.1M	1.22	0.13
ResNet-34 [2] *	5.7M	1.95	0.212
ECAPA-TDNN (512) [6]	6.2M	1.01	0.127
ECAPA-TDNN (1024) [6]	14.7M	0.87	0.107
S-vector + PLDA [9]	25.3M	2.67	0.30
BC-CMT-Tiny	273.6K	2.70	0.254
BC-CMT-Small	1.4M	1.05	0.088
BC-CMT-Base	6.3M	0.86	0.073

with the top 1,000 imposter selections were applied after cosine similarity scoring. The results were evaluated in terms of the equal error rate (EER) and minimum detection cost function (minDCF) with target probabilities of 0.01 and 0.05.

5. Results and analysis

Tables 2 and 3 summarize the performance of the proposed models, and compare them with state-of-the-art models in the VoxCeleb-1 trials.

When comparing models with similar sizes, the proposed BC-CMT-Tiny model moderately outperformed the Julien *et al.* [24] and ECAPA-TDNNLite [25]; the latter two are lightweight versions of the QuartzNet [28] and ECAPA-TDNN, respectively. Although the ECAPA-TDNNLite was trained under the supervision of a large model [6] using a knowledge distillation method, the relative performance improvements in EER were 18.4% and 12.1%, respectively. Concurrently, the BC-CMT-Small achieved significant relative improvements of 51.6% and 80.7% compared with the Fast-ResNet-34 [26] and SAEP [8], respectively; these models were CNN- and Transformer-based representative speaker embedding extractors, respectively. The BC-CMT-Small also outperformed D-TDNN-SS [5], a densely connected TDNN-based model, achieving relative improvements of 32.3% in minDCF_{0.01} with 54.8% fewer parameters. Finally, our proposed BC-CMT-Base exhibited overall superior performance compared with the ECAPA-TDNN [6] with an embedding size of 512 and ResNet-34, using a similar number of model parameters. Comparatively, the BC-CMT-Base achieved lower EER and minDCF_{0.01} scores than the ECAPA-TDNN [6] with an embedding size of 1,024, and outperformed the S-vector [9] with PLDA scoring [29], which substituted the TDNN structure of the X-vector [1] with the MHSA of the Transformer [7]. In summary, all results suggest that our proposed BC-CMT architecture successfully combined CNNs and Transformer models by adopting CMT [14] and BRL [15],

Table 4: Ablation study of proposed BC-CMT architecture evaluated in terms of EER(%) and minDCF_{0.05}.

No.	Systems	# Params	VoxCeleb-1 O	
			EER	minDCF
	BC-CMT-Small	1.4M	1.05	0.061
A.1	w/o MSPA	1.0M	1.23	0.078
A.2	+ w/ ASP [6]	984.1K	1.68	0.101
B.1	w/o BC-LMHSA	1.6M	1.56	0.095
B.2	w/o BC-IRFNN	1.5M	1.63	0.097
B.3	w/o BRL	1.6M	1.88	0.110

while efficiently maintaining the model sizes.

Table 4 investigates the effect of the proposed structural modifications on the proposed speaker embedding extractor architecture. The experiments regarding temporal pooling mechanisms and BRL techniques are labeled as A and B, respectively. Note that, without using MSPA, the proposed FS-ASP (A.1) was compared with the pooling mechanism of [6] (A.2). First, both experiments, A.1 and A.2, demonstrate the effectiveness of the MSPA. Moreover, when comparing A.1 and A.2, the EER and minDCF_{0.05} of the latter were increased by 36.6% and 29.5%, respectively, relative to the former. This indicates that employing frequency-axis weighted first- and second-order statistics for ASP is beneficial for speaker modeling. Second, without adopting the proposed BC-IRFNN (B.1) or BC-LMHSA (B.2), the overall performance was significantly degraded. Moreover, when the BC-IRFNN and BC-LMHSA were not used (B.3), the EER and minDCF_{0.05} increased by 79.0% and 80.3%, respectively, with an increment of 14.3% in the number of model parameters. These results show that broadcasting frequency-aware temporal features, which were extracted using dimension-wise convolutions, with a residual connection is effective for building a CMT-based speaker embedding extractor.

6. Conclusions

We proposed a novel speaker embedding extractor architecture that employs the BRL techniques in the LMFS and IRFFN modules of CMT. Moreover, we proposed FS-ASP, which is a temporal pooling mechanism that utilizes frequency-axis weighted statistics of input feature maps for speaker embedding. The experimental results suggest that parameter-efficient speaker embedding extractors with improved TI-SV performances can be implemented by adopting the proposed methods.

7. Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2021-0-00456, Development of Ultra-high Speech Quality Technology for Remote Multi-speaker Conference System)

8. References

- [1] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.
- [2] W. Cai, Z. Cai, X. Zhang, X. Wang, and M. Li, "A novel learnable dictionary encoding layer for end-to-end language identification," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 5189–5193.
- [3] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.
- [4] S. 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, Feb. 2021.
- [5] Y.-Q. Yu and W.-J. Li, "Densely connected time delay neural network for speaker verification," in *Proc. INTERSPEECH*, 2020, pp. 921–925.
- [6] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. INTERSPEECH*, 2020, pp. 3830–3834.
- [7] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 3830–3834.
- [8] P. Safari, M. India, and J. Hernando, "Self-attention encoding and pooling for speaker recognition," in *Proc. INTERSPEECH*, 2020, pp. 941–945.
- [9] N. J. M. S. Mary, S. Umesh and S. V. Katta, "S-vectors and TESA: speaker embeddings and a speaker authenticator based on transformer encoder," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 404–413, 2022.
- [10] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. INTERSPEECH*, 2018, pp. 2252–2256.
- [11] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Proc. Odyssey*, 2018, pp. 74–81.
- [12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. International Conference on Learning Representations (ICLR)*, 2021.
- [13] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing convolutions to vision transformers," in *IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 22–31.
- [14] J. Guo, K. Han, H. Wu, C. Xu, Y. Tang, C. Xu, and Y. Wang, "CMT: Convolutional neural networks meet vision transformers," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12175–12185.
- [15] B. Kim, S. Chang, J. Lee, and D. Sung, "Broadcasted residual learning for efficient keyword spotting," in *Proc. INTERSPEECH*, 2021, pp. 4538–4542.
- [16] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: a largescale speaker identification dataset," in *Proc. INTERSPEECH*, 2017, pp. 2616–2620.
- [17] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. INTERSPEECH*, 2018, pp. 1086–1090.
- [18] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv:1606.08415*, 2020.
- [19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. International Conference on Machine Learning (ICML)*, 2021, pp. 22–31.
- [20] S. Chang, H. Park, J. Cho, H. Park, S. Yun, and K. Hwang, "Subspectral normalization for neural audio data processing," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021, pp. 850–854.
- [21] S. Seo, D. J. Rim, M. Lim, D. Lee, H. Park, J. Oh, C. Kim, and J.-H. Kim, "Shortcut connections based deep speaker embeddings for end-to-end speaker verification system," in *Proc. INTERSPEECH*, 2019, pp. 2928–2932.
- [22] F. Wang, J. Cheng, W. Liu, , and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, July, 2018.
- [23] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conference on Learning Representations (ICLR)*, 2015.
- [24] J. Balian, R. Tavarone, and M. P. A. Coucke, "Small footprint text-independent speaker verification for embedded systems," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021, pp. 6179–6183.
- [25] Q. Lin, L. Yang, X. Wang, X. Qin, J. Wang, and M. Li, "Towards lightweight applications: Asymmetric enroll-verify structure for speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2022.
- [26] 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," in *Proc. INTERSPEECH*, 2020, pp. 2977–2981.
- [27] Z. N. Karam, W. M. Campbell and N. Dehak, "Towards reduced false-alarms using cohorts," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 4512–4515.
- [28] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020, pp. 6124–6128.
- [29] S. Ioffe, "Probabilistic linear discriminant analysis," in *Proc. European Conference on Computer Vision (ECCV)*, 2006, pp. 531–542.