JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 40, 303-316 (2024)
DOI: 10.6688/JISE.202403_40(2).0007

Speaker Verification System Based on Time Delay Neural Network with Pre-activated CNN Stem and Deep Layer Aggregation

WEI-TING LIN, TING-WEI CHEN AND CHIA-PING CHEN+
Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung, 804 Taiwan
E-mail: {m093040020; m103040017}@student.nsysu.edu.tw; cpchen@mail.cse.nsysu.edu.tw

In this paper, we improve the state-of-the-art ECAPA-TDNN model for speaker verification with CNN stem, self-calibration (SC) block, and deep layer aggregation. The proposed architecture is called Emphasized Channel Attention Propagation and Deep Layer Aggregation with Pre-activated CNN Stem in Time Delay Neural Network, which is abbreviated as ECAPDLA CNNv2-TDNN. First, we add a pre-activated stemming convolution layer in front of the main ECAPA-TDNN architecture. This ensures that the input to our main model architecture is a stable feature representation. Next, we change the multi-layer aggregation of ECAPA-TDNN to deep layer aggregation and replace the SE-Res2block in ECAPA-TDNN with SC block. Thus, the proposed implementation enhances feature extraction on multiple time scales and spectral channels and improves the overall training efficacy. On the VoxCeleb1-O dataset, the proposed model achieves an equal error rate (EER) of 0.95%. This is significantly better than the EER of 1.23% achieved by the ECAPA-TDNN baseline.

Keywords: speaker verification, time delay neural network, CNN stem, self-calibration block, deep layer aggregation

1. INTRODUCTION

Protecting personal data from unauthorized usage has become increasingly imperative in modern digital world. Automatic authentication via biometric feature extraction and verification, such as face recognition and fingerprint verification, can be used to protect data security. Automatic authentication systems have been used in personal services and deployed on personal devices. As alternative to faces or fingerprints, voiceprints (biometrics extracted from acoustic/speech signals) are natural, convenient, fast, and accurate. Authentication based on voiceprints is called speaker verification.

In this paper, we focus on speaker verification. In a nutshell, the purpose of speaker verification is to decide whether the test utterance and the specified enrollment data belong to the same person. A very successful approach to speaker verification is based on Gaussian mixture models (GMM) [1]. Subsequently, a front-end factor analysis speaker verification technique called *i*-vector [2] achieves better performance than GMM. With the breakthrough of deep learning, many speaker verification systems based on deep neural networks (DNNs) can outperform systems based on *i*-vectors. Today, state-of-the-art

Received July 22, 2022; accepted April 13, 2023.
Communicated by Berlin Chen.

+ Corresponding author.

speaker verification systems are based on time delay neural networks (TDNN) called x-vectors [3]. A recent successful speaker verification model in the mainstream TDNN methodology is ECAPA-TDNN [4], which is built with multi-branch topology [5] and 1-D convolution neural network (CNN). ResNet [6], a popular image recognition model with 2-D CNN, is shown to be effective in speaker verification. This leads to the proposal of ECAPA CNN-TDNN [7], which combines ECAPA-TDNN and a stemming 2-D convolution layer. The added stemming layer allows features of audio to be reconstructed and retains significant information before input to the TDNN layer. The hybrid structure further promotes the performance of speaker verification systems.

Inspired by the ECAPA CNN-TDNN model, we propose the following refinements on baseline ECAPA-TDNN models. First, we add a stem layer in front of ECAPA-TDNN. Different with ECAPA CNN-TDNN, we replace the residual block with pre-activated residual block. This step of preprocessing audio can effectively improve the model's performance. In addition, the pre-activated residual block can effectively maintain the stability of training. Next, we replace the aggregation in ECAPA-TDNN with deep linear aggregation (DLA) [8] to efficiently extract information at different resolutions. We also replace the SE-Res2Blocks in ECAPA-TDNN with SC blocks [9]. An SC block is a structure with multi-branch topology and self-attention mechanism. We name the proposed model ECAPDLA CNNv2-TDNN model. It outperforms the baseline ECAPA-TDNN model on the VoxCeleb1 dataset [10].

The remainder of this paper is organized as follows. Section 2 describes the architecture of our speaker verification system. Section 3 describes our proposed method and model. Section 4 introduces datasets and training protocol. Section 5 discusses and analyses the experimental results. Finally, Section 6 provides some concluding remarks.

2. SPEAKER VERIFICATION SYSTEM

Fig. 1 shows the architecture of our proposed speaker verification system, which includes four stages: Data processing and augmentation, Feature extraction model, Feature comparison, and Performance evaluation. The following subsections describe the details of each stage.

2.1 Data Processing and Augmentation

In the data processing and augmentation stage, we process the data to make it suitable for training. Our audio file format uses wav files. If the data is in other formats, *e.g.*, sph, m4a, we convert it to wav files using sph2pipe or ffmpeg tools. After that, we extract 200 or 400 frames from the audio files as input for training, and apply several data augmentation methods to the data.

Data augmentation is helpful for neural network training that requires a lot of data. It not only saves the trouble of labeling additional data, but also increases the amount and diversity of data. We use MUSAN [11] and RIR [12] for data augmentation. The MUSAN corpus contains three parts: speech, music and noise. The RIR corpus is the reverberation produced by the reflection of periodic impulse sound in room environments. At each training stage, one of the above additive or convolutional noises is randomly selected for data augmentation.

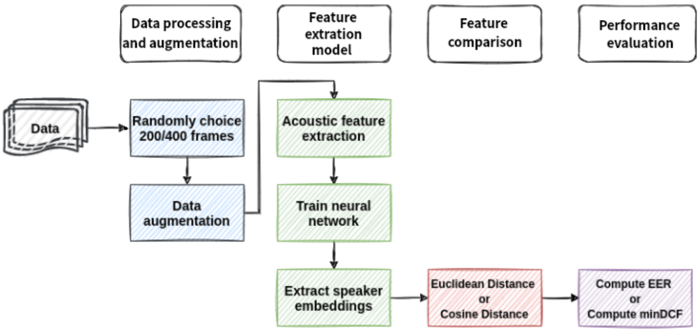
Data processing and augmentation
Feature extration model
Feature comparison
Performance evaluation
Data
Randomly choice 200/400 frames
Data augmentation
Acoustic feature extraction
Train neural network
Extract speaker embeddings
Euclidean Distance or Cosine Distance
Compute EER or Compute minDCF

Fig. 1. The architecture of our speaker verification system. There are four stages: Data processing and augmentation, Feature extraction model, Feature comparison, and Performance evaluation.

2.2 Feature Extraction Model

In the feature extraction model stage, we first extract acoustic features from the frames processed in the previous stage. The Mel-Frequency Cepstral Coefficients (MFCC) use Discrete Cosine Transform (DCT) which may lose some nonlinear information. In recent speaker verification competitions, most teams prefer to use the filter bank that preserves the essence of the sound signal. Thus, we use filter banks as our acoustic features. Next, we use the acoustic features as input to train the neural network. Neural network architecture is one of the key points of deep learning methods. A well-designed architecture can effectively improve the ability to extract speaker embeddings. There are two types of frameworks commonly used in speaker verification. One is a time-delay neural network based on 1-D CNN, and the other is a residual neural network based on 2-D CNN. We adopt the mainstream model ECAPA-TDNN based on TDNN, and use it as the benchmark model for improvement and training. Our proposed model will be discussed in the Materials and Method section. The last step in feature extraction model is to extract speaker embeddings. The trained neural network model acts as an extractor, extracting features that represent the speaker.

2.3 Feature Comparison

In the feature comparison stage, the speaker embedding is used to calculate the Euclidean distance or cosine distance to determine the similarity of the two audio segments. There are three main reasons for using these two methods for backend scoring. First, when there is no significant difference between the training data and the test data, the use of cosine similarity or Euclidean distance is still robust. Second, since most of the current state-of-the-art models apply angle-based loss functions, cosine similarity or Euclidean distance also performs well on these loss functions. Third, compared with other feature comparison methods, cosine similarity or Euclidean distance greatly improves the efficiency of the operation.

2.4 Performance Evaluation

In the performance evaluation stage, the scores are adjusted and calculated through the threshold to obtain evaluation criteria such as Equal Error Rate (EER) and Minimum

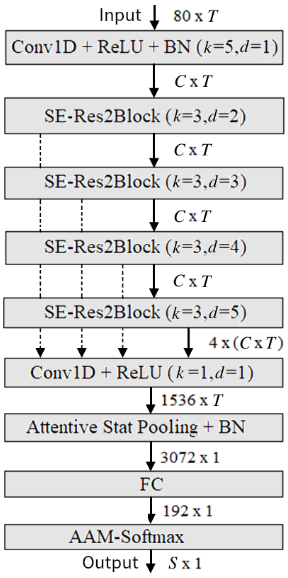
Input
80 x T
Conv1D + ReLU + BN (k=5,d=1)
C x T
SE-Res2Block (k=3,d=2)
C x T
SE-Res2Block (k=3,d=3)
C x T
SE-Res2Block (k=3,d=4)
C x T
SE-Res2Block (k=3,d=5)
4 x (C x T)
Conv1D + ReLU (k =1,d=1)
1536 x T
Attentive Stat Pooling + BN
3072 x 1
FC
192 x 1
AAM-Softmax
Output
S x 1

Fig. 2. The architecture of ECAPA-TDNN. The parameter T is the number of input frames, C is the number of convolution channels, k is the size of the convolution kernel, d is the dilation rate, and S is the number of speakers.

Decision Cost Function (minDCF), which are used to evaluate the performance of the speaker verification system. In speaker verification systems, the main error cases are False Rejection (FR) and False Acceptance (FA). Equal error rate makes the sum of the two error rates at a minimum. Another evaluation method is the minDCF. Detection Cost Function (DCF) is the weighted sum of the ratios of false alarms and misses. Here, we set the parameters P_{target}, $C_{\text{false alarm}}$, and C_{miss} of minDCF as 0.01, 1, and 1, respectively.

3. MATERIALS AND METHOD

3.1 ECAPA-TDNN

We use the ECAPA-TDNN model as the baseline model for improvement. ECAPA-TDNN[4] is a model based on TDNN and won the first place in the VoxSRC-20 competition. The architecture of ECAPA-TDNN is shown in Fig. 2. The input will be an 80-dimensional acoustic feature vector multiplied by T frames. The first layer of the ECAPA-TDNN model is Conv1D, ReLU, and BatchNormalization. Then, there will be 4 layers of 1-D Squeeze-Excitation Res2Block (SE-Res2Block). Each layer of SE-Res2Block adopts different dilation rates, which are 2, 3, 4, and 5 respectively. The next layer is Conv1D and ReLU, which is used for Multi-layer feature aggregation and summation. Combine the outputs of SE-Res2Block with different dilation rates in the previous layer. The next layer is the Attentive Statistical Pooling layer, which computes the weighted mean and weighted standard deviation. Then, the fully connected layer and the BatchNormalization layer linearly transform the features to get the speaker embedding.

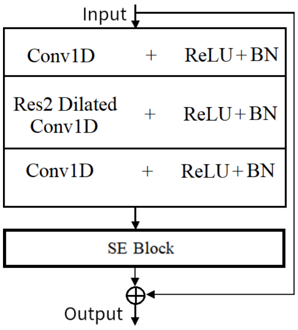

Fig. 3. The architecture of SE-Res2Block; It consists of Res2Block module and SE block.

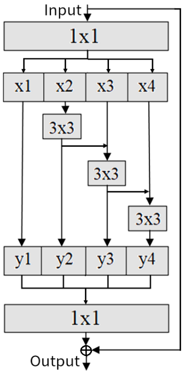

Fig. 4. The architecture of Res2Block.

The last layer is AAM-Softmax [13], which calculates the probability of belonging to each speaker.

The most significant part of the ECAPA-TDNN model is SE-Res2Block. The architecture of SE-Res2Block is shown in Fig. 3. SE-Res2Block adds SE Block [14] to the end of Res2Block module [5]. SE is an acronym for compression and excitation. The SE block enables the model to generate an attention-like learning mechanism for each layer of channels. In addition, the original 2-dimensional convolution is changed to 1-dimensional convolution with dilation rate. The architecture of Res2Block is shown in Fig. 4. The feature is divided into four groups of x1, x2, x3, and x4 (the feature can be divided into any number of groups, here we take four groups as an example). x1 is passed directly to y1. x2 is passed to y2 through a convolutional layer with a kernel size of 3. And it is also used as the input to the convolutional layer of x3. x3 and the output of the previous stage (x2) are passed through the convolutional layer to y3 and used as the input to the x4 convolutional layer. x4 and the output of the previous stage (x3) are passed to y4 through the convolutional layer. After these steps, y1, y2, y3, y4 are concatenated and fed into a 1×1 convolutional layer to integrate the collected features. In ECAPA-TDNN, there are only 4 layers of SE-Res2Block. Although the ECAPA-TDNN architecture is not complicated, it has excellent performance in the field of speaker verification. Many subsequent studies are developed on the basis of ECAPA-TDNN.

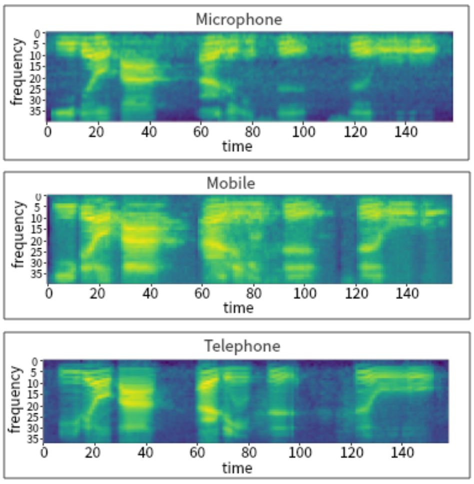

Fig. 5. The spectrogram of the same utterance by the same speaker on different devices. From top to bottom are the spectrograms of the microphone, mobile, and telephone. The vertical axis is the frequency axis, and the horizontal axis is the time axis.

3.2 2D Convolution Stem

By presenting the sound signal in the audio file as a spectrogram, it can be found that there is still a slight bias under almost the same conditions. And these biases have the potential to interfere with the predictions of the neural network model. We plotted the spectrogram of the same utterance by the same speaker on different devices, as shown in Fig. 5. We found that despite the same speaker and the same utterance, there are still slight shifts on the frequency axis and the time axis. And these shifts are more obvious on the frequency axis than time axis. Therefore, we refer to [7] to solve the problem of the shift on the frequency axis. [7] uses a 2D CNN to perform convolution operations on the input features before applying the TDNN main architecture. The architecture of the 2D CNN is simple as shown in Fig. 6. The first layer is Conv2D, ReLU and BatchNormalization. After that, there are two layers of residual block. Then another layer of Conv2D, ReLU and BatchNormalization. The number of channels C in CNN stem is set to 64 for all experiments. In addition, the stride of the first and final 2D convolution layers is set to 2 in the frequency dimension. The output feature maps of CNN stem would be flattened to match the input of ECAPA-TDNN network. 2D CNN builds the input features into a local, frequency-invariant feature before being fed into the main architecture. In this paper, we refer to this 2D CNN simply as CNN to distinguish it from subsequent research.

[15] mentioned that changing the order of the residual block in CNN can make training easier and enhance the generalization capacity of the model. [15] indicates that if the activation function is an identity mapping, the information can be transferred between the residual units in forward or backward propagation. To create such an identity mapping, the activation function (ReLU and BN) on the original information path is changed from the traditional "Post-Activation" to "Pre-Activation". The architecture of the original residual block and the pre-activated residual block is shown in Fig. 7. There are two effects of pre-activation. The first is Ease of optimization, which makes training easier be-

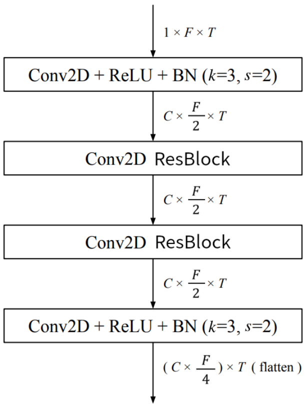
$1 \times F \times T$
Conv2D + ReLU + BN (k=3, s=2)
$C \times \frac{F}{2} \times T$
Conv2D ResBlock
$C \times \frac{F}{2} \times T$
Conv2D ResBlock
$C \times \frac{F}{2} \times T$
Conv2D + ReLU + BN (k=3, s=2)
$(C \times \frac{F}{4}) \times T$ (flatten)

Fig. 6. The architecture of CNN. The kernel size denoted by k, and stride denoted by s. C, F, T correspond to the channel, frequency dimension, and temporal dimension, respectively.

cause the activation function is identity mapping. The second is Reducing overfitting. Using BN as pre-activation is equivalent to regularizing the model. Through pre-activation, training becomes easier and more efficient. In this paper, we refer to this modified CNN simply as CNNv2 to distinguish it from previously described architecture.

3.3 Deep Layer Aggregation

Feature extraction through layers of neural networks may lead to loss of information. The issue of information loss may be alleviated by feature aggregation to preserve information appropriately. Aggregation methods may focus on aggregating features at different spatial resolutions or different depths. Among aggregation methods, iterative deep aggregation (IDA) iteratively aggregates the features extracted from different time scales through different depths. Hierarchical deep aggregation (HDA) aggregates the features of blocks in a tree-structured architecture.

Deep layer aggregation (DLA) [8] combines the advantages of IDA and HDA to aggregate time and channel feature information effectively. The architecture of DLA is shown in Fig. 8. There are four stages in DLA. The first stage has two layers of residual blocks and aggregation nodes that aggregate the above two layers. In the second stage, there are four layers of residual blocks and two aggregation nodes with different numbers of inputs. In the third stage, its architecture is similar to combining the two second-stage architectures. However, the last aggregation node in the third stage aggregates more features. In the last stage, there is the same architecture as the first stage. Finally, the outputs of the aggregation nodes in the last stage can be passed through pooling and fully connected layers to obtain the speaker embeddings. From the first stage to the final stage, DLA iteratively merges feature representations from different time scales to achieve iterative deep aggregation. That is, the output of the last aggregation node of each stage will be passed backwards. In addition, DLA aggregates channel features of different depths through a tree-structure to achieve hierarchical deep aggregation.

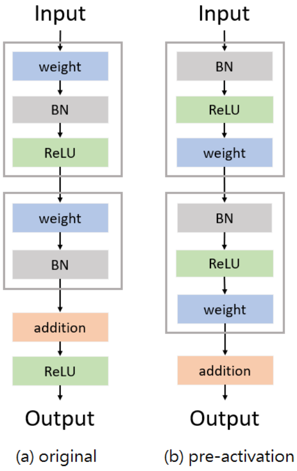

Fig. 7. (a) is the architecture of the normal residual block. The BN layer is connected after the weight layer. All ReLUs are connected after BN except the last ReLU which is connected after the addition; (b) is the pre-activated residual block architecture. Place BN and ReLU before the weight layer.

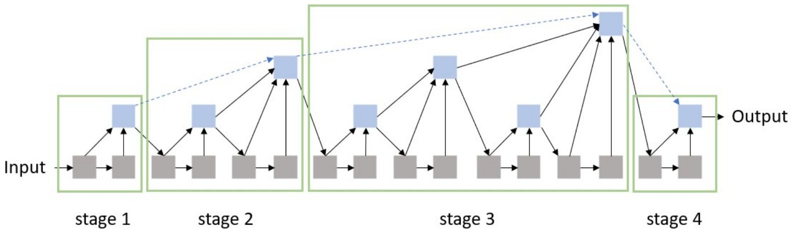

Fig. 8. The architecture of DLA. The grey blocks represent residual blocks. The blue blocks represent aggregation nodes. The green boxes represent different stages, each of which implements hierarchical deep aggregation. The blue dotted line passes the aggregated features to the aggregation nodes of the subsequent stage as input, implementing iterative deep aggregation.

3.4 ECAPDLA CNNv2-TDNN

We propose the ECAPDLA CNNv2-TDNN model based on the above materials. The architecture of ECAPDLA CNNv2-TDNN is shown in Fig. 9. Essentially, we add CNNv2 stem and deep layer aggregation on top of ECAPA-TDNN. ECAPDLA CNNv2-TDNN consists of two parts, CNNv2 and ECAPDLA-TDNN. In the CNNv2 part, the input is 80-dimensional log mel-filterbanks. The first and last layers are Conv2D, ReLU and batch normalization (BN). The middle two layers are both pre-activated residual blocks. With the CNNv2 architecture, we construct a frequency-invariant, stable, and easily trainable input feature for the subsequent ECAPDLA-TDNN.

In the ECAPDLA-TDNN part, consider that ECAPA-TDNN uses only the basic residual block of Res2Net. We refer to [16] to replace it with SC block, the convolutional block of Self-Calibration Network (SCNet) [9], which integrates the concepts of Res2Net and SKNet [17]. The structure of SC block is shown in Fig. 10. Similar to Res2Net bottle-

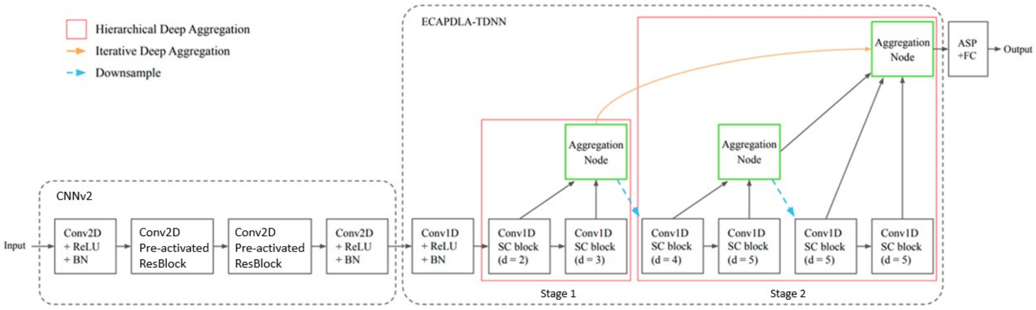

Fig. 9. The architecture of ECAPDLA CNNv2-TDNN. It consists of CNNv2 and ECAPDLA-TDNN.

neck residual block (Res2Block), SC block divides input feature maps into several groups. The difference is that SC blocks use self-attention mechanisms instead of only hierarchical residual-like connections to capture spatial information. The mechanism is called self-calibration. The self-calibration mechanism generates additional spaces of different scales through average pooling (AvgPool), convolution, and FC, and multiplies with the original scale space. With self-calibration, the model with SC block can exploit features at multiple scales and avoid spurious signals through the attention mechanism. Based on the above characteristics of SC block, we propose to use it to replace SE-Res2Block.

Furthermore, we adopt DLA to replace the multi-layer aggregation in ECAPA-TDNN. Note the complete DLA architecture is too big to match the size of ECAPA-TDNN. Instead, only the first half of DLA is implemented. ECAPDLA-TDNN begins with a 512-dimensional layer with Conv1D, ReLU and BN. Subsequently, the flow of signal propagates through stage 1 and stage 2, with a total of 6 SC blocks and 3 aggregation nodes. In stage 1, there are 2 SC blocks with dilation rates of 2 and 3, respectively. Additionally, there is an aggregation node to aggregate these two SC blocks. The output of the aggregation node will be downsampled to 512 to maintain consistency. In addition, the output of the aggregation node will be used as the input of stage 2, as well as passed to the last aggregation node of stage 2. In stage 2, there are 4 SC blocks with dilation rates of 4, 5, 5, and 5, respectively. Here, ECAPDLA-TDNN is similar to ECAPA-TDNN with an increased dilation rate. However, ECAPDLA-TDNN limits the dilation rate to 5 to avoid losing information continuity. The architecture of stage 2 is similar to combining two stage 1s. The first half of stage 2 is an aggregation node that aggregates two layers of SC blocks, and its output is also downsampled to 512. The special thing about the second half of stage 2 is that the input of the last aggregation node consists of two SC blocks, the aggregation node of the first half, and the aggregation node of the previous stage. This aggregation node aggregates feature representations at different levels and stages. The output of the last aggregation node of stage 2 will be set to 1,536 dimensions for the following attention statistics pooling layer (ASP) and fully connected layer (FC). Finally, we will get a 192-dimensional speaker embedding from FC layer. With speaker embeddings, we can calculate how similar the registered and test speakers are by Euclidean distance or cosine distance.

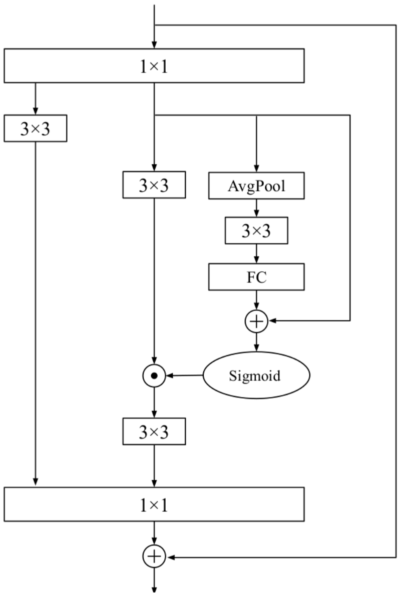

Fig. 10. The structure of SC block. The convolution structure with branch topology and sigmoid function on the right part of SC block is the self-calibration convolution. The '+' symbol denotes element-wise summation. The '•' symbol denotes element-wise multiplication.

4. EXPERIMENTS

4.1 Datasets

Our systems use the development of VoxCeleb2 (VoxCeleb2 Dev) [18] as the training set. VoxCeleb2 Dev is a text-independent dataset and consists of 1,092,009 utterances from 5,994 speakers. VoxCeleb2 Dev extracts audio clips from Youtube videos and covers speakers of different ages, accents, and ethnicities. The sampling rate of audio signals is 16 kHz and the language is English.

Our models are evaluated on the VoxCeleb1 dataset (cleaned), including VoxCeleb1-O, VoxCeleb1-E, and VoxCeleb1-H. Their detailed information is shown in Table 1. VoxCeleb1-O is the most basic dataset with a smaller number of speakers and test pairs. VoxCeleb1-E is an extension of VoxCeleb1-O with a higher number of speakers and test pairs. VoxCeleb1-H is more challenging than the above two datasets because the speakers in each pair are of the same country and gender.

Table 1. Detailed information on VoxCeleb1-O, VoxCeleb1-E, and VoxCeleb1-H datasets.

Dataset	number of speakers	number of utterances	number of pairs
VoxCeleb1-O	40	4,708	37,611
VoxCeleb1-E	1,251	145,160	579,818
VoxCeleb1-H	1,190	137,924	550,894

4.2 Training Protocol

We downsample the sampling rate of audio to 8 kHz. The audio waveform is converted to 80-dimensional Mel-filterbank energies with a window length of 25 ms and a frame-shift of 10-ms . In addition, the extracted features are mean-normalized without the application of voice activity detection (VAD). All models are trained with an initial learning rate of 0.001, which is reduced by 25% every 10 epochs with Adam optimizer [19]. The setting of AAM-softmax is margin of $m = 0.2$ and pre-scaling of $s = 30.0$. A weight decay of 2e-05 was applied during training to prevent the model from overfitting. The batch size for training is set to 128. All ECAPA-TDNN architectures consistently have 512 channels in convolution frame layers. The channel of extended stem structure is set to 64. The scale dimension s in Res2Block is set to 8. The kernel size and stride r for AvgPool in SC block is set to 4. The output size of speaker embedding is 192.

5. RESULTS AND DISCUSSION

5.1 Comparison of Stem Structures

Here we compare different stem structures used in extending ECAPA-TDNN models. The experimental results of different CNN stems are shown in the first to third rows of Table 2. ECAPA-TDNN is the baseline model. ECAPA CNN-TDNN is to add the CNN stem of the original residual block to the front of ECAPA-TDNN. ECAPA CNNv2-TDNN is a model that changes the residual block in ECAPA CNN-TDNN to pre-activation. It can be seen that ECAPA CNNv2-TDNN is better than the above two models in each dataset of VoxCeleb1. The preprocessed CNN stem structure is to construct frequency-invariant features that ignore small and reasonable shifts in the frequency domain to compensate for actual speaker frequency variations. Both CNN and CNNv2 use 2D convolutional layers in the beginning and last part of stem, only the residual blocks are different. The original residual block is a simple operation where a tensor passes through two convolution layers, then a short-cut tensor is added. Thus, CNN stem helps the model to learn robust frequency-invariant feature representation. The pre-activated residual block places BN and ReLU before the weight layer. It enables direct transfer of information between residual units, making the model easier to train. The identity mapping achieved by pre-activation reduces the probability of overfitting during training. Pre-activating residual blocks reinforces the effect of creating stable feature representation in front of the main architecture.

Table 2. Experimental results of all models.

Model	VoxCeleb1-O		VoxCeleb1-E		VoxCeleb1-H	
	EER(%)	minDCF	EER(%)	minDCF	EER(%)	minDCF
ECAPA-TDNN	1.2392	0.1585	1.4290	0.1641	2.6292	0.2545
ECAPA CNN-TDNN	1.1488	0.1402	1.4057	0.1586	2.5795	0.2461
ECAPA CNNv2-TDNN	1.0424	0.1274	1.2646	0.1454	2.3518	0.2230
ECAPDLA-TDNN	1.1701	0.1464	1.3281	0.1525	2.4767	0.2389
ECAPDLA CNN-TDNN	1.1807	0.1214	1.2484	0.1487	2.3053	0.2317
ECAPDLA CNNv2-TDNN	**0.9520**	**0.1086**	**1.1332**	**0.1370**	**2.2055**	**0.2167**

5.2 Comparison of Aggregation Methods

Here we compare different aggregation methods. We experiment with multi-layer aggregation (MLA) and the proposed deep layer aggregation (DLA). As can be seen from the first and third rows of Table 2, DLA outperforms MLA in all experiments. The advantages of DLA over MLA are as follows. First, DLA integrates more feature information from different receptive fields with more aggregation nodes than MLA. Secondly, DLA aggregates feature information at different depths to better capture feature details, while MLA concatenates information of the same depth. Lastly, DLA recursively integrates feature information from different time scales, so it extracts speaker information over various time scales and spectral channels. As a result, DLA outperforms MLA on ECAPA-TDNN models in our tests.

5.3 Comparison of All Models

In this subsection, we combine CNN stem and DLA to compare with all the previously mentioned models. Table 2 are the experimental results of all models. ECAPDLA CNN-TDNN changes the aggregation method of ECAPA CNN-TDNN to DLA and adopts SC block. ECAPDLA CNNv2-TDNN is the model proposed in this study, which is introduced by Section 3.4. First, it can be seen that our proposed ECAPDLA CNNv2-TDNN model outperforms all other models. ECAPDLA CNNv2-TDNN builds a robust feature representation through CNNv2. The DLA aggregation method then extracts features at different time resolutions and depths. In addition, the SC block enhances the discriminativeness of the speaker embedding. Through these methods, the ECAPDLA CNNv2-TDNN model can effectively extract speaker embedding to classify speakers. Next, we observe the experimental results on the VoxCeleb1-O dataset. It can be found that the scores of each model are not much different, and the model with CNN stem has a better score. This is because on the simpler dataset (VoxCeleb1-O), building robust features and using simple models can also achieve successful results. Finally, it can be seen that the model with DLA achieves better scores on the more challenging VoxCeleb1-E and VoxCeleb1-H datasets. This is because with the strong feature extraction capabilities of DLA and SC block, speakers who are difficult to distinguish can be better classified.

6. CONCLUSION

In this paper, we modify the ECAPA-TDNN model by adding a CNN stem preprocessing and using deep layer aggregation (DLA) with different receptive fields. In addition, we improve the training performance by adjusting the order of residual blocks in the CNN stem. Finally, we use SC block to replace Res2Block of ECAPA-TDNN to improve the ability to extract features. Overall, the proposed ECADLA CNNv2-TDNN model outperforms ECAPA-TDNN model baseline by 23.1% on VoxCeleb1-O. The above study will serve as the basis for our future efforts on improving state-of-the-art methods for speaker verification based on ECAPA-TDNN. Furthermore, we will experiment with other challenging datasets to fully realize the advantages of ECADLA CNNv2-TDNN.

ACKNOWLEDGMENT

We thank the Department of Computer Science and Engineering, National Sun Yat-Sen University for its research environment and computational resources. We are grateful to colleagues in our laboratory for the advice to solve system bugs and correct grammatical errors in our write-up.

REFERENCES

1. D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, Vol. 10, 2000, pp. 19-41.
2. F. Curelaru, "Front-end factor analysis for speaker verification," in *Proceedings of International Conference on Communications*, 2018, pp. 101-106.
3. D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: robust DNN embeddings for speaker recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5329-5333.
4. B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," *arXiv Preprint*, 2020, arXiv:2005.07143.
5. 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, 2019, pp. 652-662.
6. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.
7. J. Thienpondt, B. Desplanques, and K. Demuynck, "Integrating frequency translational invariance in TDNNs and frequency positional information in 2D ResNets to enhance speaker verification," *arXiv Preprint*, 2021, arXiv:2104.02370.
8. F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2403-2412.
9. J.-J. Liu, Q. Hou, M.-M. Cheng, C. Wang, and J. Feng, "Improving convolutional networks with self-calibrated convolutions," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 096-10 105.
10. A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv Preprint*, 2017, arXiv:1706.08612.
11. D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv Preprint*, 2015, arXiv:1510.08484.
12. 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 *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 5220-5224.
13. J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690-4699.

14. J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132-7141.
15. K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proceedings of European Conference on Computer Vision*, 2016, pp. 630-645.
16. 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," in *Interspeech*, 2021, pp. 76-80.
17. X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 510-519.
18. J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv Preprint*, 2018, arXiv:1806.05622.
19. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv Preprint*, 2014, arXiv:1412.6980.

Wei-Ting Lin received the BS degree in Department of Computer Science and Information Engineering from Tatung University in 2020 and the MS degree in Computer Science and Engineering from National Sun Yat-sen University in 2022. His research interests include multimedia information technology and artificial intelligence.

Ting-Wei Chen received the BS degree in Department of Computer Science and Information Engineering from Feng Chia University in 2021. He is currently pursuing MS degree in Computer Science and Engineering from National Sun Yat-sen University. His research interests include machine learning and automatic speaker verification.

Chia-Ping Chen received the BS degree in Physics from National Taiwan University, the MS degree in Physics from National Tsing-Hua University, and the MS and Ph.D. degrees in Electrical Engineering from the University of Washington at Seattle. Since 2005, he has been a member of the faculty of the Department of Computer Science and Engineering of National Sun Yat-sen University, Kaohsiung, Taiwan. His research interests mainly focus on spoken language technology and applications, including speech recognition, speech synthesis, speaker recognition, and acoustic sound event detection.