# A Speaker Recognition Method Based on Dynamic Convolution with Dual Attention Mechanism

Yuan Luo, Kuilin Zhu, Wenhao Wang, Ziyao Lin

**Abstract — Deep neural networks have gained significant attention in text-independent speaker recognition tasks. However, due to the fixed parameters of traditional static convolutional neural networks, they cannot flexibly capture the variation in phonemes that are integral to speech sentences. To address this limitation, this paper proposes a channel-space attention-based dynamic convolutional speaker recognition method. This method employs dual-attention mechanisms to generate dynamic convolutional kernels, which improves the capture of phoneme variation information between different inputs in the speech signal. We conducted experiments using the TIMIT dataset to evaluate the proposed method's effectiveness in various network frameworks. Our results show that the best performance can be achieved when dynamic convolution is generated using four static convolutional kernels. Specifically, in the ResNet-34 framework, the Equal Error Rate (EER%) of the proposed method is improved by 31.1% over the static convolutional method CNN and by 20.3% over the single-attention dynamic convolutional method (DynamicConv). Additionally, the performance of the proposed method is enhanced in all other network frameworks. These findings demonstrate the effectiveness of the proposed method and the importance of considering phoneme variations in speaker recognition systems.**

***Index Terms*—Speaker Recognition, Deep Learning, Attention Mechanism, Dynamic Convolution**

Manuscript received November 7, 2022; revised April 25, 2023. This work was supported in part by the National Natural Science Foundation Youth Fund Project (Grant No. 61703067), the Chongqing Basic Science and Frontier Technology Research Project (Grant No. Cstc2017jcyjAX0212), and the Chongqing Municipal Education Commission Science and Technology Research Project (KJ1704072).

Yuan Luo is a professor of School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, P.R. China (e-mail: 1739705031@qq.com)

Kuilin Zhu is a postgraduate student of School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, P.R. China (Corresponding author, phone: 166-382-41148; e-mail: 1104569847@qq.com)

Wenhao Wang is a postgraduate student of School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, P.R. China (e-mail: a1104569847@163.com)

Ziyao Lin is a postgraduate student of School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, P.R. China (e-mail: S200431307@stu.cqupt.edu.cn)

## I. Introduction

The speaker recognition task aims to identify the speaker based on their speech signal. The key to this problem lies in extracting the speaker's identity from the speech signal. Initially, solutions for the speaker recognition task employed GMM-UBM, support vector machines, and joint factor analysis [1-3] for acoustic modeling, and the best performance was obtained using the GMM-UBM/i-vector approach. However, with the emergence of deep learning, DNN-UBM/i-vector [4] models that use DNN instead of GMM have shown better results. Inspired by this, many researchers have modeled speaker recognition tasks as classification tasks using deep learning networks to obtain speaker identity features such as d-vector, x-vector [5-6], etc. Speaker feature recognition networks typically follow a common architecture, beginning with preprocessing the acoustic signal to extract acoustic features that align more closely with human auditory perception. Commonly used acoustic features include MFCC features and Log Mel spectrum [7-8]. The acoustic features are then input into a deep neural network to extract frame-level features for the speaker. These frame-level features are subsequently pooled, with options such as mean pooling, maximum pooling, and pooling with fused attention, to obtain segment-level features[9-10]. Finally, the segment-level features are classified by a fully connected layer to obtain the final speaker identity features. The extraction of speaker frame-level features is a critical element in speaker recognition, and researchers have investigated various network architectures, such as ResNet, TDNN, CNN-LSTM, and BLSTM-ResNet [12-15], to enhance the extraction of these features.

Speech text is composed of phonemes and acoustic features, and text-independent feature extraction is required for text-independent speaker recognition. However, recent studies have shown that the performance of deep learning-based feature extraction networks is significantly affected by phonemes [16-18], highlighting the importance of phonemes in text-independent speaker recognition tasks. Traditional static convolutional neural networks have fixed parameters and may struggle to capture the variability in phonemes across different speech inputs, which can limit the performance of speaker recognition networks. The Adaptive Convolutional Neural Network (ACNN) [19-21] provides a solution to the issue of fixed convolutional kernel parameters by allowing the network

model to dynamically adjust them based on the input. This model has been widely applied in computer vision [22-24] and natural language processing [25-26], yet current research on its use in speaker recognition remains insufficient [27-28]. Dynamic convolution is a type of ACNN that captures varying phoneme information in different input data by fusing information from multiple static convolutions through an attention mechanism and dynamically adjusting the convolution kernel parameters based on the input data. While current dynamic convolutional neural networks are primarily used for image classification tasks [29], channel attention is an appropriate method to generate the weight matrix of the dynamic convolutional kernel. However, since phoneme information in speech recognition tasks is more evident in the frequency domain, it is insufficient to rely solely on channel attention to capture the richer information on phoneme variation.

Based on the analysis presented above, this paper proposes a novel dynamic convolutional network for speaker recognition, called CSDA-DCNN, which utilizes a dual attention mechanism based on channel and spatial attention [30-31]. The network replaces the traditional static convolution method with dynamic convolution to address the issue of fixed parameters after training, which limits the network's ability to adapt to phoneme changes in different speakers' speech. Furthermore, the proposed dual attention mechanism of channel and spatial fusion improves the limitations of single-attention dynamic convolutional networks that only use the channel attention mechanism. This new approach is more effective for speech processing tasks and can capture richer information on phoneme change.

This paper is organized as follows: Part II presents the proposed CSDA-DCNN network approach, Part III provides a detailed description of the experimental procedure and data analysis, and Part IV summarizes the contributions of this paper.

## II. Proposed Method

The proposed CSDA-DCNN method employs a dual-attention mechanism that combines the benefits of both channel and spatial attention mechanisms, in order to generate dynamic convolution kernels that capture the phonetic characteristics of different speakers. This results in speaker identity features that contain richer information and lead to improved accuracy in the speaker recognition network. The overall framework of the model is illustrated in Figure 1.

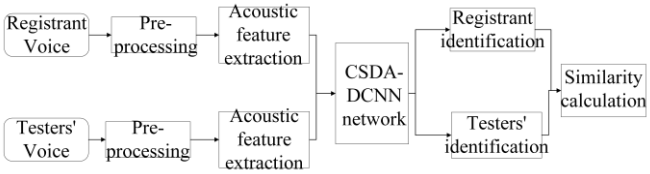



Fig.1. Overall flow of speaker recognition

The overall framework of the speaker recognition model proposed in this paper comprises four main modules, namely the pre-processing module, the acoustic feature extraction module, the network module, and the scoring judgment module.

The pre-processing module aims to prepare the speaker's input speech signal for further analysis. It includes signal denoising, pre-emphasis, framing and windowing, and valid speech detection.

The acoustic feature extraction module is responsible for extracting effective feature parameters from the pre-processed speech signal. As the sampled speech signal is a one-dimensional signal in the time domain, it needs to be processed using time-domain or frequency-domain analysis methods to extract feature parameters that can effectively represent the sound information.

The network module utilizes the proposed CSDA-DCNN convolutional network to learn the speaker identity features. The model parameters are updated using a loss function, and the final extracted feature vector is matched to the speaker's identity.

Finally, the scoring module computes the similarity between the extracted speaker identity feature vectors using methods such as cosine similarity to determine the identity of the speaker.

### A. *Network*

In this study, ResNet34[32] was chosen as the baseline framework and was improved by replacing its convolutional layers with the proposed dual-attention dynamic convolution module, as illustrated in Fig 2. The CSDA-DCNN block is the core of the proposed dual-attention dynamic convolution method, and it has the same structure in Layer2, Layer3, and Layer4 as in Layer1, except for the parameters of the convolution kernel in each CSDA-DCNN block.

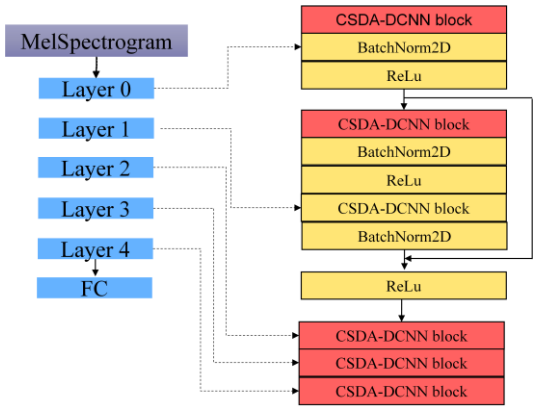



Fig.2. The speaker recognition network framework proposed in this paper

The parameters of the CSDA-DCNN modules in each layer and the input and output dimensions are shown in Table I.

TABLE I
THE NETWORK PARAMETERS OF THE PROPOSED CSDA-DCNN

| Layer | Input-Size | Output-Size | Kernel | Stride |
|---|---|---|---|---|
| Layer0 | 256×1×40×300 | 256×16×20×300 | Conv2d(1,16,7) | 2×1 |
| Layer1 | 256×16×20×300 | 256×16×20×300 | Conv2d(16,16,3) | 1×1 |
| Layer2 | 256×16×20×300 | 256×16×10×150 | Conv2d(16,32,3) | 2×2 |
| Layer3 | 256×16×10×150 | 256×64×5×75 | Conv2d(32,64,3) | 2×2 |
| Layer4 | 256×64×5×75 | 256×128×5×75 | Conv2d(64,128,7) | 2×1 |

### B. *Dual Attention Mechanism Module*

Inspired by the human attentional mechanism, individuals do not necessarily see every pixel of an entire image at once when viewing an image, but rather focus their attention on specific parts of the image based on their needs. Additionally, humans learn where to direct their attention when viewing an image in the future based on their prior experiences with images. In deep learning, the attention mechanism is a mechanism whereby the network learns a set of weighting coefficients and dynamically adjusts them to emphasize areas of interest while suppressing irrelevant background areas.

*1) Channel Attention*

The channel attention mechanism is a commonly used method for implementing attention in deep learning. It involves assigning different weights to data from different channels to emphasize important features and suppress irrelevant ones. This is achieved by adding an attention weight parameter to each channel. Figure 3 illustrates the implementation of this method.

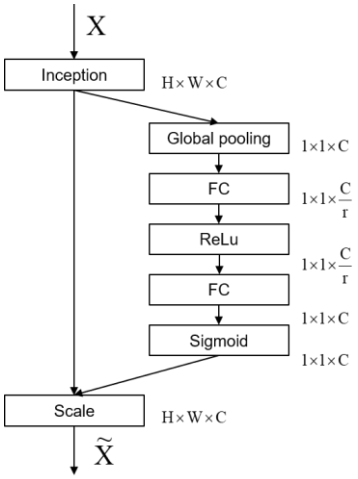



Fig. 3. Channel attention network model

The input data is initially pooled using either the maximum or average value method, which reduces the dimensionality from $C \times H \times W$ to $C \times 1 \times 1$, with each channel having only one corresponding number. It is then passed through a fully connected layer and normalized to the range of 0 to 1 using the Softmax layer. The resulting data can be interpreted as the weight of each channel, which is then used to determine the degree of emphasis that each channel has on the final output. In this paper, the proposed model improves upon the standard channel attention mechanism by removing the original Softmax layer and normalizing the data after it has been added to the corresponding positions of the spatial attention matrix, as illustrated in Figure 4.

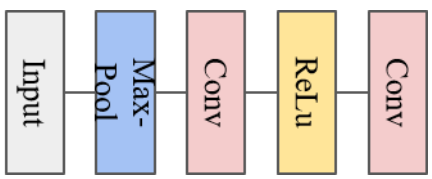



Fig4 Modified channel attention network model

The channel attention module is calculated as follows:

$$M_{avg} = conv(c_{in,}k,1)(AvgPool(x)) \quad (1)$$

Where $X$ is the input data, which is pooled globally averaged and then convolved to obtain the attention weight parameter for each channel $M_{avg}$. Parameter $c_{in}$ is the input channel of the data, $k$ is the number of output channels, which is also the number of static convolution kernels that need to be set in the final method in this paper, and 1 is the convolution kernel size.

The final matrix of channel attention parameters $M_{channel}$ is obtained by activating matrix $M_{avg}$ with the ReLu function and then convolving it:

$$M_{channel} = conv(k,k,1)(\mathrm{Re}\,LU(M_{avg})) \quad (2)$$

Where $k \times k$ is the input and output channels of the convolution kernel and 1 is the convolution kernel size.

*2) Spatial Attention*

The spatial attention mechanism ends up with a weight matrix of dimension $1 \times H \times W$, which corresponds to the weights at each position in space. The input data are pooled in the dimension of the channel for the maximum value as well as the mean value. The pooled data is reduced from dimension $C \times H \times W$ to dimension $2 \times H \times W$ and then convolved to dimension $1 \times H \times W$. The calculation process is shown in Figure 5.

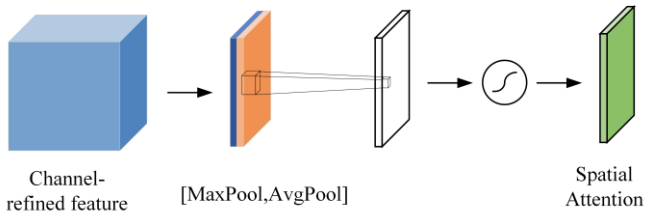



Fig5 Spatial attention network model

The model proposed in this paper modifies the spatial attention mechanism. After calculating the spatial attention parameter with dimension $1 \times H \times W$, it is then flattened by the Flatten operation and then convolved to obtain the final spatial attention weight parameter. This is shown in Figure 6.

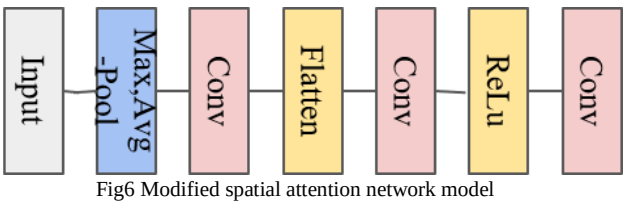



Fig6 Modified spatial attention network model

The spatial attention module is calculated as follows.

The dimension of the input data is reduced to $2 \times H \times W$ after $MaxPool$ and $AvgPool$ and then to $1 \times H \times W$ by convolution:

$$M_s = conv(c_{in},1,1)(MaxPool, AvgPool(X))) \quad (3)$$

Where the $M_s$ matrix is the matrix of weight parameters for each position on the space.

The $M_s$ matrix is flattened by the Flatten operation to a dimension of $b \times T$ and $b$ is the batch size of the input data, after which the convolution operation is performed.

$$M_F = conv(T, k, 1) Flatten(M_s) \quad (4)$$

Where $M_F$ is the length of the data after spreading, $k$ is the number of static convolutional kernels and 1 is the convolutional kernel size.

The $M_F$ matrix is activated by the ReLu function and then convolved to obtain the final spatial attention weight parameter matrix $M_{spatial}$.

$$M_{spatal} = conv(k, k, 1)(\mathrm{Re}\, Lu(M_F)) \quad (5)$$

Where $k \times k$ is the input and output channels of the convolution kernel and 1 is the convolution kernel size.

*3) Dual Attention Mechanism*

The network model for the dual-attention mechanism module is illustrated in Figure 7, which mainly focuses on the calculation of the channel and spatial attention weight parameter matrices for the input data, and the utilization of these matrices as the fusion matrix for the dynamic convolution kernel.

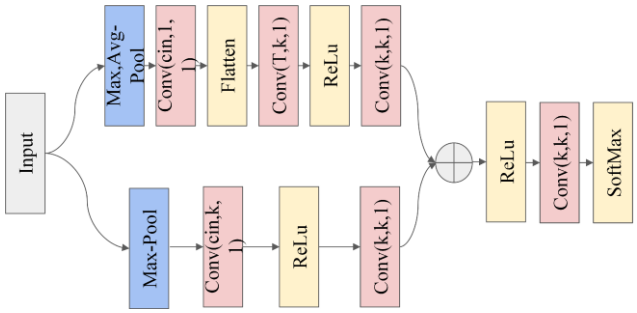



Fig.7 Dual attention mechanism network model

After inputting the data, the channel attention weight matrix and the spatial attention weight matrix are calculated respectively, and the elements are summed to obtain $M_{cs}$.

$$M_{cs} = M_{channel} \oplus M_{spatial} \quad (6)$$

Where $M_{channel}$ is the channel attention weight matrix and $M_{Spatial}$ is the spatial attention weight matrix. The final dynamic convolutional attention weight $M_{attention}$ is obtained by convolving $M_{cs}$ and then normalizing the output data to the interval 0 to 1 by SoftMax.

$$M_{attention} = SoftMax(conv(k, k, 1)(M_{cs})) \quad (7)$$

Where $k \times k$ is the input and output channels of the convolution kernel and 1 is the convolution kernel size.

*C. CSDA-DCNN Network*

The attention mechanism used in the DynamicConv[33] approach is GAP+FC+ReLu+FC+Softmax, also known as squeeze-and-excitation (SE Net)[34]. The dynamic convolution module is shown in Fig 8.

This paper replaces its single attention approach with the dual attention mechanism proposed in this paper.

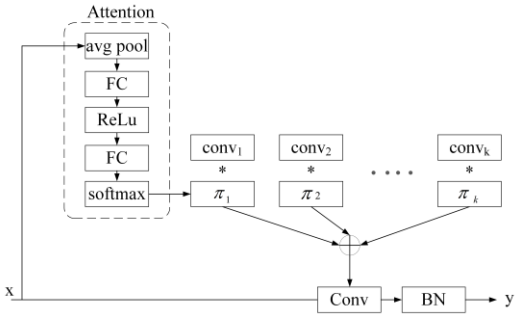



Fig. 8. Dynamic convolution module

The dual-attention dynamic convolution block designed in this paper is shown in Fig 9. It can be seen that the key aspect of the method proposed in this paper is the use of a dual-attention mechanism to generate dynamic convolutional kernels. The input data shown in the figure are dimension of $X \ni R^{C_{in} \times H \times W}$ and the input data are extracted for channel and spatial attention separately. The channel attention weight parameter extracted by the channel attention module is of dimension $b \times k$, $b$ is the batch size of the input data. $k$ is the number of static convolution kernels in the dynamic convolution calculation module. For spatial attention, the data is first pooled for maximum value and pooled for mean value, and then convolved to obtain a spatial attention weight matrix of dimension $1 \times H \times W$. This data is then subjected to a Flatten operation, and in order to obtain the final spatial attention parameter, the flattened data is convolved and its dimension reduced to $b \times k$. Finally, the weight matrices obtained from the channel attention and spatial attention modules are summed up at the corresponding positions and then normalised to the range 0 to 1 by the Softmax layer, i.e. the final attention weight parameter matrix $M_{attention}$ is obtained.

The dynamic convolution module uses the dual attention weight matrix calculated in the previous stage to weight and fuse multiple static convolution kernels, and the parameter matrix of the dynamic convolution kernel is calculated as

$$\tilde{W} = \sum_{k=1}^{K} \pi_k(x) \tilde{W}_k \quad (8)$$

Where $\pi_k$ is the weight assigned to each static convolutional kernel and is the parameter matrix for each static convolutional kernel. The bias of the dynamic convolution kernel is calculated as:

$$\tilde{b} = \sum_{k=1}^{K} \pi_k(x) \tilde{b}_k \quad (9)$$

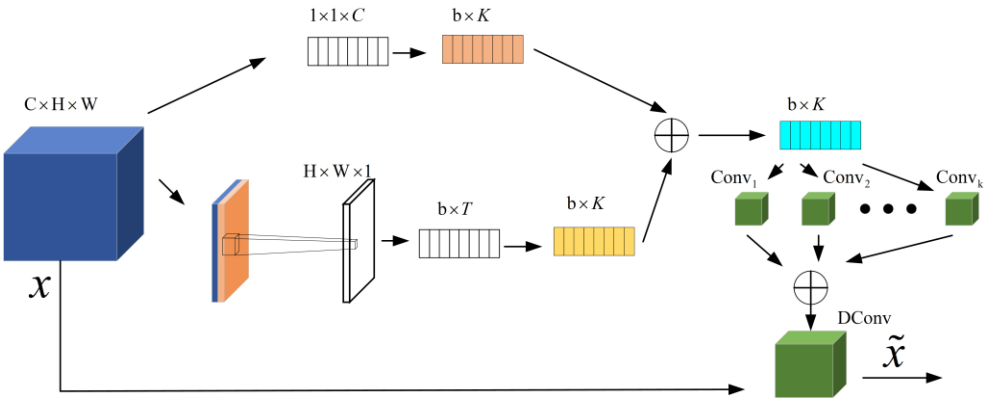



Fig.9 Dual-attention dynamic convolution block

Where $\pi_k$ is the weight assigned to each static convolutional kernel and $\tilde{b}_k$ is the bias weight of each static convolutional kernel.

The final dynamic convolution kernel is calculated as

$$y = g(W^T x + b) \tag{10}$$

Where $W^T$ is the parameter matrix of the dynamic convolution kernel and $b$ is the bias of the dynamic convolution kernel.

## III. Experimental Results and Analysis

### A. Dataset

To verify the effect of phonemes on speaker recognition, the TIMIT dataset[35], which is rich in phoneme information, is used for this experiment. The dataset captures speech information from eight major dialect regions in the United States, with a sampling frequency of 16kHz and a total of 6300 speech items. All speech data were manually segmented and labeled at the phoneme level (phone level). These include 3150 sentences with compact phonemes and 1890 sentences with divergent phonemes.

### B. Experimental Details

*1) Pre-processing*

Convolutional neural networks are widely used in image recognition tasks, where the input is typically two-dimensional data. However, speech signals are one-dimensional signals that vary over time. Therefore, to apply convolutional methods to speech signal processing tasks, it is necessary to first convert the one-dimensional speech signal into a two-dimensional signal. Various methods can be used for this purpose, such as Meier spectrum and MFCC. In this paper, we choose the Mel spectrum as the input signal for the neural network, as it is more informative. The processing process of the Mel spectrum is shown in Figure 10. Pre-processing the input signal into a two-dimensional format enables the convolutional neural network to capture information in the spatial range without losing information in the temporal domain. The window length is 25ms, the step length is 10ms, and the extracted data dimension is 40. STFT, which stands for Short-Time Fourier Transform, is used to calculate the frequency spectrum of the signal and its energy spectrum.

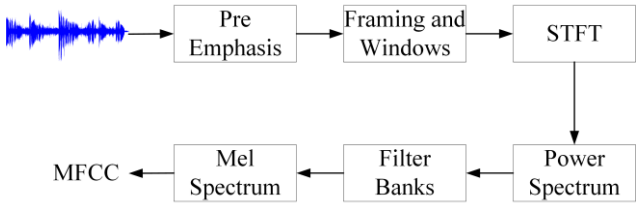



Fig 10. Mel spectrum generation process

*2) Back-end Scoring and Evaluation Metrics*

In the training phase, the deep learning model is trained using a fully connected layer for speaker identification as a classification task. The trained model has the ability to extract the identity features of the speaker from the speaker's audio information to distinguish between different speakers. In the validation phase, the final fully-connected layer of the deep learning model is removed, and the output vector before the fully-connected layer is obtained as the identity features of that speaker. Then, the similarity calculation is performed to evaluate the performance of the speaker recognition model. There are various similarity calculation methods available such as Cosine Similarity, PLDA, Two Covariance PLDA [36-38], among others. However, for the purpose of this paper, the relatively simple cosine similarity calculation method is uniformly used as the scoring method. The similarity score can be calculated using the following formula.

$$similarity = \cos(\theta) = \frac{A \bullet B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{11}$$

Experiments use equal error rate as an evaluation criterion. The mean error rate is a common evaluation metric for speaker recognition tasks and can be derived from the following equation.

$$FRR = \frac{Num(FN)}{Num(TP) + Num(FN)} \tag{12}$$

$$FAR = \frac{Num(FP)}{Num(FP) + Num(TN)} \quad (13)$$

$$EER = FRR = FAR \quad (14)$$

Where $FRR$ is the false rejection rate, which is the error caused by misidentifying the target speaker as a non-target speaker. $FAR$ is the false acceptance rate is the error caused by discriminating a non-target speaker as the target speaker. For a particular voice recognition system, with the error acceptance rate as the horizontal coordinate and the error rejection rate as the vertical coordinate, the error rate is called the equal error rate (EER) by adjusting the threshold so that the values of the error acceptance rate and the error rejection rate are equal. Obviously, the smaller the value of the equal error rate, the better the system performance.

*3) Experimental Environment*

The experimental environment is based on the Pytorch framework and is trained on an NVIDIA RTX2080ti graphics card without using any data enhancement methods. The experimental operating system is ubuntu 18.04.

### C. Experimental Results and Analysis

*1) Static Convolutional Kernel Number Testing*

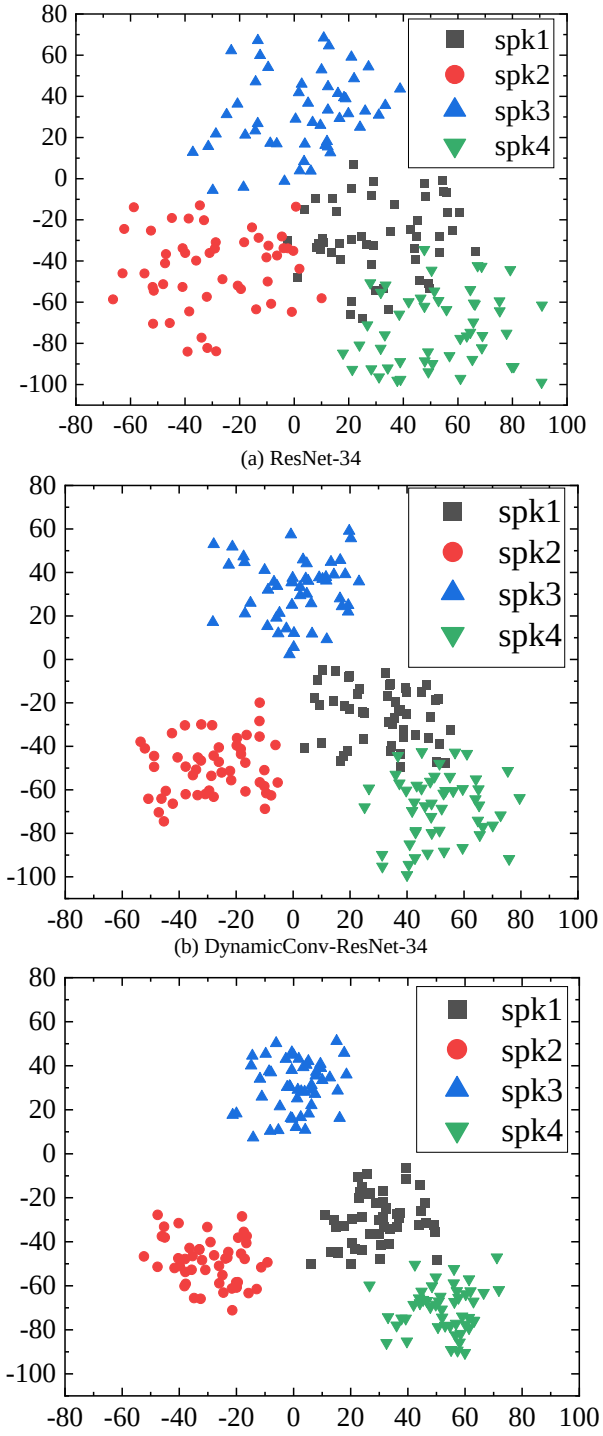
Since the number of static convolutional kernels has a strong relationship with the complexity and accuracy of the model, this paper uses CSDA-DCNN convolutional networks for experiments with different numbers of convolutional kernels. The experimental results are shown in Table II. The best performance can be obtained when the number of convolutional kernels K=4, and the performance decreases when K=6. Therefore, the number of convolutional kernels K=4 is chosen for the subsequent experiments and analysis.

TABLE II
COMPARATIVE ANALYSIS OF THE NUMBER OF STATIC CONVOLUTION KERNELS IN THE RESNET-34 FRAMEWORK

| Literature | Method | convolutional kernel number | EER% |
|---|---|---|---|
| ResNet-34 | CSDA-DCNN | 2 | 2.21 |
| ResNet-34 | CSDA-DCNN | 4 | 1.88 |
| ResNet-34 | CSDA-DCNN | 6 | 1.96 |
| ResNet-34 | CSDA-DCNN | 8 | 2.09 |

*2) Speaker Embedding Clustering Visualization Test*

In order to visualize the superiority of the proposed method in this paper, we use ResNet-34, DynamicConv-ResNet-34, CSDA-DCNN-ResNet-34, respectively, and the output before the final fully connected layer of the network as the identity feature of this speaker, and use the t-SNE [39] algorithm to project the speaker identity feature vector down to the two-dimensional plane, and its visualization is shown in Fig 11. A total of four speakers with 50 sentences each were used in the experiment, for a total of two hundred sentences. The figure shows that all four speaker identity features of the network using CSDA-DCNN-ResNet-34 are well aggregated, while the speaker identity features of the network using DynamicConv-ResNet-34 as well as ResNet-34 are all relatively farther away from the center.




(a) ResNet-34

(b) DynamicConv-ResNet-34

(c) CSDA-DCNN-ResNet-34

Fig.11. Visualization of speaker embedding clustering

*3) Performance Test of ResNet-34 Network under Different Convolutional Methods*

In order to verify the effectiveness of the proposed dual-attention dynamic convolution method, this paper compares it with two other convolution methods: DynamicConv and CSDA-DCNN, as well as ResNet-34, which does not use dynamic convolution. The performance of these methods is evaluated on the training set, and the results are shown in Figure 12. The CSDA-DCNN approach achieved the best performance among the three methods.

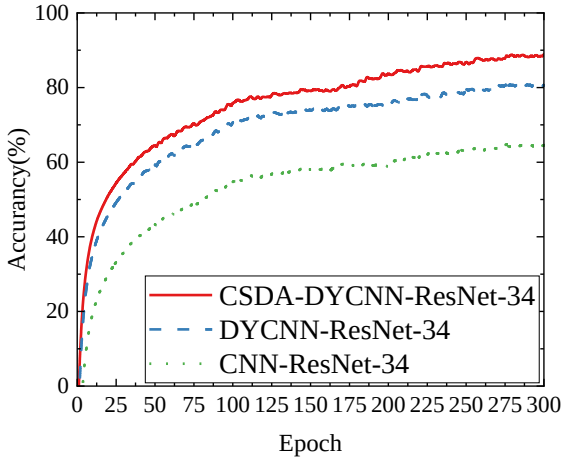


Fig.12. Module training performance

The experimental results on the validation set are presented in Table III. The results indicate that the proposed method in this paper outperforms ResNet-34 with traditional static convolution by 31.1%, and the traditional dynamic convolution method (DynamicConv) by 20.3% in terms of accuracy. These findings demonstrate that the dynamic convolution method utilizing the dual-attention mechanism is more effective in capturing phoneme information and improving the performance of the speaker recognition system.

TABLE III
COMPARISON OF THE ACCURACY USING THE PROPOSED CSDA-DCNN, AND DYNAMICCONV, CNN IN THE RESNET-34 FRAMEWORK.

| Literature | Method | EER% |
|---|---|---|
| ResNet-34 | - | 2.73 |
| ResNet-34 | DynamicConv | 2.36 |
| ResNet-34 | CSDA-DCNN | 1.88 |

*4) CSDA-DCNN Method Adaptation Test*

To verify the applicability of the proposed dual-attention dynamic convolution method in different network frameworks, VGG-M[40],is used as the framework and CSDA-DCNN is used as the convolution layer. The experimental results on the validation set are shown in Table IV.

TABLE IV
COMPARISON OF THE ACCURACY USING THE PROPOSED CSDA-DCNN, AND DYNAMICCONV, CNN IN THE VGG-M AND RESNET-34 FRAMEWORK.

| Literature | Method | EER% |
|---|---|---|
| ResNet-34 | - | 2.73 |
| ResNet-34 | DynamicConv | 2.36 |
| ResNet-34 | CSDA-DCNN | 1.88 |
| VGG-M | - | 4.12 |
| VGG-M | DynamicConv | 3.78 |
| VGG-M | CSDA-DCNN | 3.26 |

The experimental results indicate that the proposed method in this paper outperforms the traditional static convolution method by 20.8% and the traditional dynamic convolution method (DynamicConv) by 13.9% in terms of accuracy when applied to the VGG-M framework. These results demonstrate the efficacy and general applicability of the dual-attention dynamic convolution method proposed in this study across different network frameworks.

*5) Comparative Analysis of the Proposed Method with Other Networks*

This paper presents a comparison of the performance of the CSDA-DCNN dynamic convolution method with previously studied methods. Additionally, the performance of the CSDA-DCNN method in several other networks is tested and presented in Table V. The experimental results show that the proposed method improves the performance of the networks to varying degrees compared to other networks. However, it still lags behind the current optimal network framework ECAPA-TDNN. One possible reason for this is that the double attention mechanism introduced in this paper increases the number of parameters, which may lead to performance degradation due to overfitting.

TABLE V
COMPARATIVE ANALYSIS OF THE PROPOSED METHOD WITH OTHER NETWORKS

| Literature | Method | EER(%) | Improvement (%) |
|---|---|---|---|
| ResNet-34 | - | 2.73 | - |
| ResNet-34 | CSDA-DCNN | **1.88** | **31.1** |
| VGG-M | - | 4.12 | |
| VGG-M | CSDA-DCNN | **3.26** | **20.9** |
| ResNet-50 | - | 4.33 | |
| ResNet-50 | CSDA-DCNN | **3.82** | **11.8** |
| Thin Res-Net-34 | - | 3.11 | |
| Thin Res-Net-34 | CSDA-DCNN | **2.69** | **13.5** |
| GMM-UBM/i-vector | - | 9.73 | |
| TDNN-UBM | - | 5.44 | |
| ECAPA-TDNN | - | **1.45** | |

## IV. CONCLUSION

In this paper, we propose a dynamic convolution method with both channel and spatial attention mechanisms to enhance the speaker recognition network. Our approach computes attention parameters from both channel and spatial perspectives, which overcomes the limitations of static convolution in capturing phoneme information and the insufficiency of traditional dynamic convolution that uses only one attention mechanism. Experiments were conducted on the TIMIT dataset, and the best results were obtained when the number of convolutional kernels was set to 4. We replaced the static convolutional kernels in ResNet-34 with CSDA-DCNN and DynamicConv for comparison experiments, and the results confirmed the effectiveness of our proposed method. However, the introduction of the dual-attention mechanism greatly increases computational effort. Therefore, future work will focus on optimizing the network structure to reduce parameters and computation. Additionally, although our method fuses channel and spatial attention, the current fusion method is a simple summation, and a more sophisticated approach should be explored in future research.

## REFERENCES

[1] Reynolds D A, Quatieri T F, Dunn R B. Speaker verification using adapted Gaussian mixture models[J]. Digital signal processing, 2000, 10(1-3): 19-41.

[2] Campbell W M, Sturim D E, Reynolds D A. Support vector machines using GMM supervectors for speaker verification[J]. IEEE signal processing letters, 2006, 13(5): 308-311.

[3] Kenny P, Boulianne G, Ouellet P, et al. Joint factor analysis versus eigenchannels in speaker recognition[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2007, 15(4): 1435-1447.

[4] Lei Y, Scheffer N, Ferrer L, et al. A novel scheme for speaker recognition using a phonetically-aware deep neural network[C]//2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2014: 1695-1699.

[5] Variani E, Lei X, McDermott E, et al. Deep neural networks for small footprint text-dependent speaker verification[C]//2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). Singapore: IEEE, 2014: 4052-4056

[6] Snyder D, Garcia-Romero D, Sell G, et al. X-vectors: Robust dnn embeddings for speaker recognition[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Barcelona: IEEE, 2018: 5329-5333.

[7] Logan B. Mel frequency cepstral coefficients for music modeling[C]//In International Symposium on Music Information Retrieval. 2000.

[8] Atal B S. The history of linear prediction[J]. IEEE Signal Processing Magazine, 2006, 23(2): 154-161.

[9] Li M Z, Zhang X L. An investigation of speaker clustering algorithms in adverse acoustic environments[C]//2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2018: 1462-1466.

[10] Wu Y, Guo C, Gao H, et al. Vector-Based Attentive Pooling for Text-Independent Speaker Verification[C]//INTERSPEECH. 2020: 936-940.

[11] Wang Z, Yao K, Li X, et al. Multi-resolution multi-head attention in deep speaker embedding[C]//ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 6464-6468.

[12] Yu Y Q, Fan L, Li W J. Ensemble additive margin softmax for speaker verification[C]//ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019: 6046-6050.

[13] Liu Y, He L, Liu J, et al. Speaker embedding extraction with phonetic information[J]. arXiv preprint arXiv:1804.04862, 2018.

[14] Jung J W, Heo H S, Yang I H, et al. Avoiding speaker overfitting in end-to-end dnns using raw waveform for text-independent speaker verification[J]. extraction, 2018, 8(12): 23-24.

[15] Zhao Y, Zhou T, Chen Z, et al. Improving deep CNN networks with long temporal context for text-independent speaker verification[C]//ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 6834-6838.

[16] Shon S, Tang H, Glass J. Frame-level speaker embeddings for text-independent speaker recognition and analysis of end-to-end model[C]//2018 ieee spoken language technology workshop (slt). IEEE, 2018: 1007-1013.

[17] Eatock J P, Mason J S. A quantitative assessment of the relative speaker discriminating properties of phonemes[C]//Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 1994, 1: I/133-I/136 vol. 1.

[18] Jung C S, Kim M Y, Kang H G. Selecting feature frames for automatic speaker recognition using mutual information[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2009, 18(6): 1332-1340

[19] Jia X, De Brabandere B, Tuytelaars T, et al. Dynamic filter networks[J]. Advances in neural information processing systems, 2016, 29.

[20] Yang B, Bender G, Le Q V, et al. Condconv: Conditionally parameterized convolutions for efficient inference[J]. Advances in Neural Information Processing Systems, 2019, 32.

[21] Chen Y, Dai X, Liu M, et al. Dynamic convolution: Attention over convolution kernels[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 11030-11039.

[22] Chen C, Ling Q. Adaptive convolution for object detection[J]. IEEE Transactions on Multimedia, 2019, 21(12): 3205-3217.

[23] Su H, Jampani V, Sun D, et al. Pixel-adaptive convolutional neural networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 11166-11175.

[24] Zamora Esquivel J, Cruz Vargas A, Lopez Meyer P, et al. Adaptive convolutional kernels[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019: 0-0.

[25] Shen D, Min M R, Li Y, et al. Learning context-sensitive convolutional filters for text processing[J]. arXiv preprint arXiv:1709.08294, 2017.

[26] Choi B J, Park J H, Lee S K. Adaptive Convolution for Text Classification[C]//NAACL-HLT (1). 2019: 2475-2485.

[27] Gu B, Guo W, Dai L, et al. An adaptive x-vector model for text-independent speaker verification[J]. arXiv preprint arXiv:2002.06049, 2020.

[28] Kim S H, Park Y H. Adaptive convolutional neural network for text-independent speaker recognition[C]//INTERSPEECH 2021. International Speech Communication Association, 2021: 641-645

[29] Chen Y, Dai X, Liu M, et al. Dynamic convolution: Attention over convolution kernels[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 11030-11039.

[30] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.

[31] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[32] Chung J S, Huh J, Mun S, et al. In defence of metric learning for speaker recognition[J]. arXiv preprint arXiv:2003.11982, 2020.

[33] Chen Y, Dai X, Liu M, et al. Dynamic convolution: Attention over convolution kernels[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11030-11039.

[34] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.

[35] Garofolo J S, Lamel L F, Fisher W M, et al. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1[J]. NASA STI/Recon technical report n, 1993, 93: 27403.

[36] Dehak N, Dehak R, Glass J R, et al. Cosine similarity scoring without score normalization techniques[C]//Odyssey. 2010: 15.

[37] Prince S J D, Elder J H. Probabilistic linear discriminant analysis for inferences about identity[C]//2007 IEEE 11th international conference on computer vision. IEEE, 2007: 1-8.

[38] Ioffe S. Probabilistic linear discriminant analysis[C]//European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2006: 531-542.

[39] Van der Maaten L, Hinton G. Visualizing data using t-SNE[J]. Journal of machine learning research, 2008, 9(11).

[40] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.