Proceedings of Machine Learning Research vol 245:1–9, 2024 2024 International Conference on Machine Learning and Intelligent Computing

SpcNet: Speaker Validation Model Based on Self-Calibrated Convolution

Xia Zhang ZHANGXIA@SXU.EDU.CN
School of Computer and Information Technology, Shanxi University

Guobo Wu WUGUOBO2021@163.COM
School of Computer and Information Technology, Shanxi University

Qian Liu STITCH0507@163.COM
School of Computer and Information Technology, Shanxi University

Editors: Nianyin Zeng and Ram Bilas Pachori

Abstract

The convolutional module-based speaker representation network has demonstrated outstanding performance in the speaker verification (SV) task and has now become one of the widely adopted network structures in this task field. There are some limitations to the convolution-based network structure, specifically with the fixed-size convolution kernel in the general convolution operation. This makes it difficult to capture long-range time-frequency and channel dependencies in speech features, limiting the network's ability to extract representations of the speaker. To overcome this issue, we have explored several alternative approaches. Firstly, we propose an enhanced self-calibrating convolutional kernel that adaptively constructs long-range time-frequency and channel dependencies around each time-frequency position. This allows for the integration of richer information, significantly enhancing the network's capacity to learn representations. Secondly, we have made adjustments to the network structure to improve the extraction of speaker feature representations. We refer to this modified model as SpcNet. In this paper, our proposed SpcNet model has been experimented on the datasets VoxCeleb1 and VoxCeleb2. Comprehensive experiments show that the Equal Error Rate (EER) is significantly improved.

Keywords: convolutional networks, long-range dependencies, self-calibrating convolution, speaker verification

1. Introduction

Voiceprint recognition, also known as Speaker Recognition, is an advanced biometric technologyand one of the products of comprehensive interdisciplinary research. Speaker Verification is one of the tasks of speaker recognition. The task aims to confirm whether a given speech signal belongs to a specific individual.

The traditional speaker validation model is represented by the The Gaussian Mixture Model (Reynolds and Rose, 1995) (GMM). Later, Reynolds et al. (2000) introduced the GMM-UBM (Gaussian Mixture Model-Universal Background Model) in the speaker verification task. In the 21st century, the i-vector (Dehak et al., 2010) model was proposed as an extension of the traditional GMM-UBM approach. It addresses the limitation of the GMM-UBM system, which assumes Gaussian components to be independent, by mapping the speaker model to a low-dimensional subspace. This improvement leads to enhanced system performance. The model maintained its status as the state-of-the-art in speaker verification for a significant period of time. In recent years,Variani et al.

were the first to train a deep neural network (DNN) to extract utterance-level speaker embedding, namely d-vectors, achieving identical performance to previous i-vector methods.

Currently, the x-vector framework has demonstrated superior performance over the i-vector method in various speaker verification tasks. Subsequently, several optimization schemes have been proposed, primarily based on the convolutional neural network (CNN) architecture, which has become the most widely used approach.ResNet stands as one of the most widely adopted network architectures for speaker verification tasks, with various variants of this architecture being extensively explored.Additionally, there exists the ECAPA-TDNN (Desplanques et al., 2020) architecture, which is based on Time Delay Neural Networks (TDNN). Currently, the ECAPA-TDNN (Desplanques et al., 2020) methods are regarded as state-of-the-art in speaker verification research. Nevertheless, ResNet has become the dominant architecture in speaker verification tasks due to its faster inference speed and satisfactory performance.

Compared to traditional speaker verification models, convolutional neural network models can effectively reduce the labor and computational overhead in classification learning tasks.However, the existing convolutional neural network models still have limitations as they can only learn similar features.Moreover, the receptive field of each time and frequency in the convolutional feature transform is primarily determined by the size of the convolutional kernel. However, the fixed-size kernel restricts the receptive field of the speech feature, limiting its ability to capture a larger context.To address the aforementioned challenges, numerous studies have proposed redesigning the fundamental convolutional modules. For example, the res2net (Gao et al., 2019) module in ECAPA-TDNN replaces the original convolutional filter with a smaller set of convolutional filter banks to expand the range of scales at which the output features can be represented. This approach aims to increase the equivalent receptive field, enabling the capture of long-range time-frequency and channel dependencies.Nevertheless, the increase in the quantity of small convolutional filters also leads to a rise in the number of parameters within the network.

In this paper, our objective is to tackle the aforementioned issues by introducing a novel approach. To accomplish this, we propose an improved self-calibrated convolutional kernel that departs from the conventional method of performing convolutional operations on a small receptive field. Our method increases the receptive field of the network, allowing each time-frequency position to adaptively capture long-range time-frequency and channel-dependent. These changes improve the recognition accuracy of the network model because only the conventional convolution operation needs to be utilized for spatial feature extraction on a segment of the input channel, eliminating redundant information in the convolutional network and reducing the number of parameters in the model. This reduction not only speeds up the computation of the network but also enhances its overall performance. In addition, we have tweaked the structure of the network to enhance its ability to extract the representation of speaker features.The experimental results verify that the method proposed in this paper is effective.

2. Approaches

2.1. Improved Self-Calibrating Convolutional Modules

Improved self-calibrating convolution enhances network performance by dynamically adjusting filter weights during learning. It divides the convolutional filter into sections, each responsible for a specific function. Assuming a set of filter banks K with shape (C,C,k_f,k_t), where C represents the channel, k_f represents frequency, and k_t represents time,it is first divided uniformly into four parts,

each responsible for a different function.Dividing the channel into two parts will result in a four-part filter, each with the dimensions ($\frac{C}{2}$,$\frac{C}{2}$,k_f,k_t).

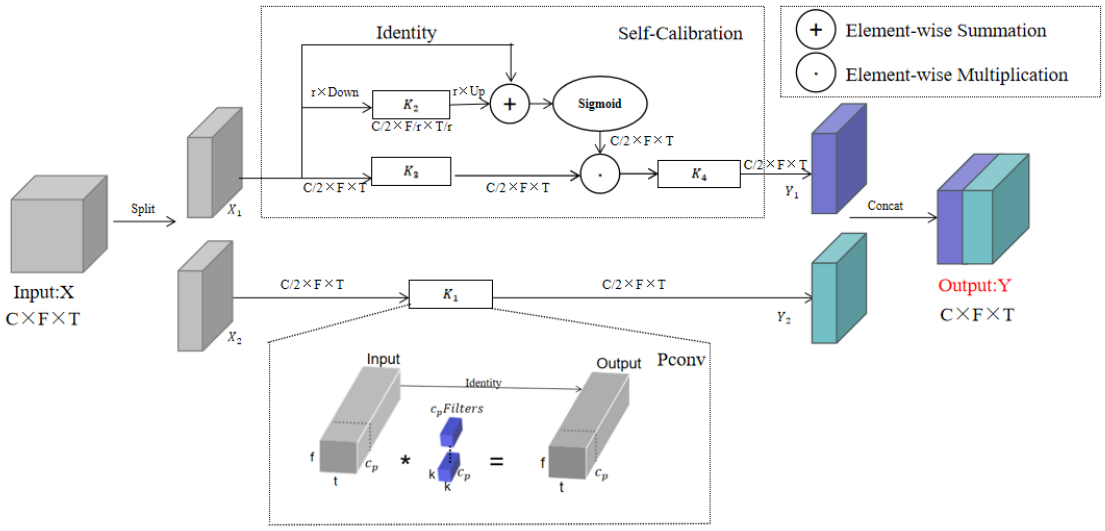

Figure 1: Improved self-calibrating convolution module.

As illustrated in Figure 1, this work divides the input X into four segments using the filter. Subsequently, the input X is evenly split into two parts, namely X1 and X2. Each part is then fed into a dedicated pathway to gather distinct wider contextual information.In the first pathway, the paper employs K1, K2, and K3 to perform a self-calibration operation on X1, resulting in Y1. Meanwhile, In another pathway, To achieve more efficient extraction of spatial features, a PConv (Chen et al., 2023) convolution operation is utilized, which reduces redundant computations and memory accesses. This approach better preserves the original spatial context and yields Y2. Moreover, this procedure reduces model computation and improves convergence speed.Finally, the intermediate outputs Y1 and Y2 are connected to form the final output, denoted as Y.

The X1 path is divided into three parts, the first of which is referred to as the ”backbone”. The backbone is responsible for extracting the fundamental features of the input data. The second part, known as the ”branching”, is responsible for extracting more advanced features from the input data. The input image is initially downsampled to obtain a small low-resolution feature map, which is then transformed by a convolution operation to acquire a compact embedding of latent space features. The third part is named the ”fusion layer”, which combines the outputs of the trunk and branches and maps the feature embeddings of the small latent space back to the original scale space. This process serves as a reference to the original feature map, guiding the feature transformation in the original scale space. This design of the self-calibrating convolution enables dynamic adjustment of filter weights during the learning process. As the basic features extracted by the backbone change, the branches can adjust their weights accordingly to extract more advanced features.

For X1 the exact operation is as follows:

Given the input X1, in this paper we use an average pool of filter size r × r and step size r as shown in the following equation:

$$\mathrm{T1} = \mathrm{AvgPool(X1)}$$

The feature transformation on T1 is performed based on K2, which is the convolutional kernel size:

$$\mathrm{X}_1^{'} = \mathrm{Up}(\mathrm{T}_1 * \mathrm{K2})$$

where Up(.) is mapping the downsampled small-scale spatial features into the original feature space. The formula is as follows:

$$Y_1^{'} = (\mathrm{X}_1 * \mathrm{K3}) \cdot \sigma(\mathrm{X1}+X_1^{'})$$

σ is the sigmoid function and "-" denotes elemental multiplication. It is useful to use as residuals in this paper to form the weights used for calibration.The final output is shown below:

$$\mathrm{Y1} = Y_1^{'} * \mathrm{K4}$$

On path X2, feature extraction is conducted using PConv, which employs convolution operation for spatial feature extraction on a segment of the input channel while keeping the remaining channel unchanged.The basic principle of the PConv method is to perform special processing on missing values in convolution operations to avoid convolution operations on missing values of input data. This method preserves valid information in the input data and concurrently diminishes the computational load and memory access requirements by limiting the convolutional operations to only a portion of the input data.

Unlike conventional convolution, the self-calibrating convolution operation adaptively treats contextual information surrounding each time-frequency position as an embedding from the latent space and models inter-channel dependencies. Rather than gathering global context, the improved self-calibrating convolution operation focuses solely on the local context surrounding each spatial location. This approach can to some extent reduce pollution information from unrelated regions. By conducting feature transformations in two scale spaces, we can better capture contextual information at different scales, thereby enhancing the model's performance and generalization capabilities.

2.2. Model Network Restructuring

In this paper, we replace the ResBlock module of the original ResNet34 with SpcBlock, as illustrated in Figure 1. Specifically, the SpcBlock first undergoes a self-calibrating convolution module with PConv, followed by normalization and activation functions, and then passes through a standard 3×3 convolution operation, as depicted in Figure 2.

In the speaker verification task, the original ResNet achieved time-frequency spatial downsampling by employing a 1×1 convolution with a stride of 2 in the propagation path. Drawing inspiration from ConvNeXt (Liu et al., 2022b) and DF-ResNet (Liu et al., 2022a), this paper introduces a separate downsampling layer comprising a BN layer and a 2×2 convolutional layer with a stride of 2. This dedicated layer is positioned after each stage, except for the last one, to achieve the same time-frequency spatial resolution downsampling as the original ResNet. Additionally, experimental evidence demonstrates that reducing the usage of activation layers actually aids in the information transfer of network features.

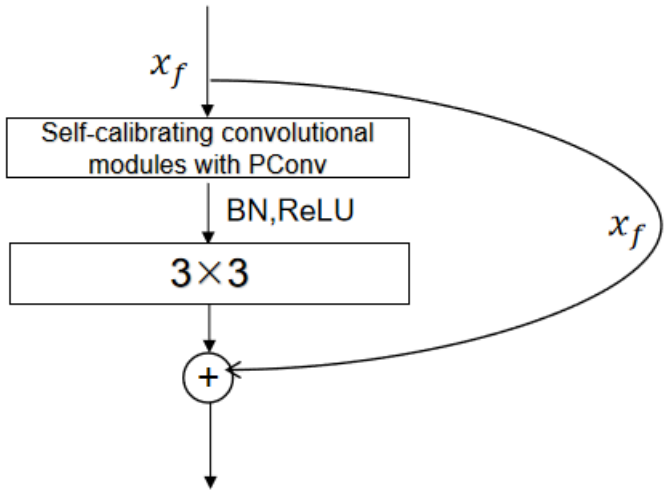

Figure 2: Spc module structure.

The overall structure of this paper is illustrated in Figure 3, and the model consists of two main modules: the frame-level feature extraction module and the utterance-level feature extraction module. The frame-level feature extraction module comprises primarily four SpcBlock modules, with a ratio of 3:3:9:3. This ratio, inspired by the descriptions in ConvNeXt and DF-ResNet, enhances the extraction of speaker utterance-level feature information.

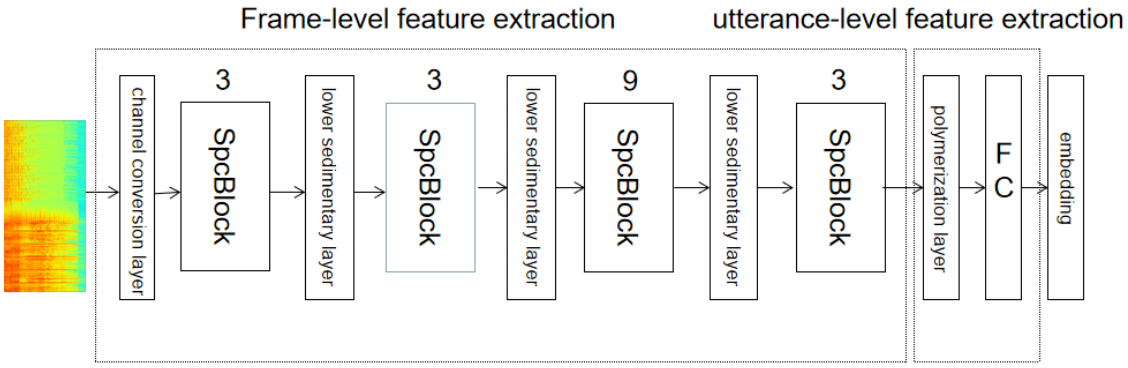

Figure 3: The overall structure of the model.

3. Experimental Setup

3.1. Datasets

To evaluate the accuracy of the proposed model for speaker verification, experiments are conducted on VoxCeleb, a widely used public dataset. The VoxCeleb dataset is divided into two parts, namely

VoxCeleb1 and VoxCeleb2. In this paper, separate test experiments are performed on these two datasets. Due to computational limitations, the ablation experiments employ the smaller dataset VoxCeleb1 as the training set and VoxCeleb1-O as the validation set. For comparative analysis with state-of-the-art models, the comparison experiments utilize VoxCeleb1-dev and VoxCeleb2-dev as the training set, while three datasets, namely VoxCeleb1-O, VoxCeleb1-E, and VoxCeleb1-H, are employed as the test set. Table 1 provides detailed information regarding the training and test sets.

Table 1: VoxCeleb dataset for training and evaluation.

Dataset	Speakers	Utterances	Trials
VoxCeleb1-dev	1,221	148,642	-
VoxCeleb2-dev	5,994	1,092,009	-
VoxCeleb1-O	40	4,715	37,611
VoxCeleb1-E	1,251	145,375	579,818
VoxCeleb1-H	1,190	138,137	550,894

3.2. Implementation Details

The network training and feature vector extraction of all models were implemented using the Pytorch deep learning framework. During training, the batch size was set to 128, and an A100 was used. The learning rate is initially set to 0.001 and decays at a decay rate of 0.02 per cycle. The models were trained using the AdamW optimizer with a weight decay of 0.05. For the loss function, AAM-softmax was employed with a scale set to 30 and a margin set to 0.2. The evaluation was performed using the equal error rate (EER) (Doddington et al., 2000) and the minimum detection cost function (minDCF) (Doddington et al., 2000) with $\mathrm{P}_{\mathrm{target}}$=0.01, C_{fa}=$\mathrm{C}_{\mathrm{miss}}$=1.

3.3. Feature Extraction and Data Enhancement

We randomly cropped a 200-frame block from an utterance for training. The model inputs consist of sixty-four-dimensional mel-filter bank features with a 25ms window size and a 10ms window shift. Data augmentation is employed to enhance data diversity, mitigate overfitting issues, and improve the model's robustness, thereby enhancing its performance and generalization. To achieve this, we utilize the noisy datasets MUSAN (Snyder et al., 2015) and RIRs (Ko et al., 2017) for data augmentation. Additionally, SpecAugment (Park et al., 2019) is applied to the final extracted log-Mel spectrogram.

4. Results

From Table 2 the EER is reduced by 5% after changing the residual block ratio. After using separate downsampling layer and reducing the activation layer EER is reduced by 9%. Then adding the improved self-calibrating convolution module EER is reduced by 17%.

Table 3 gives a comparison of the results of this paper's model with models such as classical ResNet and the ECAPA-TDNN model for the small dataset VoxCeleb1, and it can be seen that the model proposed in this paper outperforms both in terms of performance. Compared to the optimal ECAPA-TDNN model EER/minDCF is improved by 8.6%/16.4%.

Table 2: Results of the proposed method in VoxCeleb1.

Method	EER(%)	minDCF
ResNet	2.70	0.3220
Changing the residual block ratio	2.57	0.3047
Individual downsampling and ReLU reduction	2.45	0.3015
SpcBlock	2.24	0.2564

Table 3: Results of different models in the VoxCeleb1 dataset.

Mould	Params	EER(%)	minDCF
Half-ResNet34	6.579M	2.70	0.3220
ResNet34	23.295M	2.63	0.3172
ResNet18	13.655M	2.64	0.3019
ECAPA-TDNN	14.729M	2.45	0.2986
SpcNet	7.481M	2.24	0.2564

Table 4: Results of different models in the VoxCeleb2 dataset.

Mould	Params	Voxceleb1-O		Voxceleb1-E		Voxceleb1-H	
		EER(%)	minDCF	EER(%)	minDCF	EER(%)	minDCF
ResNet34SE	23.599M	1.26	0.1264	2.09	0.2548	3.53	0.3420
ECAPA-TDNN	14.729M	1.13	0.1780	1.29	0.1557	2.49	0.2633
SpcNet	7.481M	1.07	0.1379	1.20	0.1264	2.32	0.1418

Table 4 gives the results of the different models on the large dataset (VoxCleeb2), with a reduction of 14.8% and 15% compared to the traditional network ResNet34 and 8.6% and 5.3% compared to the ECAPA-TDNN model.

Table 5: Self-calibrating the convolution module in the VoxCeleb1 ablation experiment.

Mould	Identity	Pooling	EER(%)	minDCF
SpcBlock	✓	-	2.36	0.2832
SpcBlock	✓	AVG	2.24	0.2564
SpcBlock	×	AVG	2.30	0.2641
SpcBlock	✓	MAX	2.33	0.2735

Table 5 presents the ablation experiments conducted on the self-calibrating convolution module in VoxCeleb1. These experiments aim to investigate the impact of different pooling sampling methods and the presence or absence of identity mapping on the model's performance. Our experiment involved adding the average pooling operator to the self-calibrating convolution module, resulting in an EER of 2.30. Additionally, when the identity map was added separately, the EER reached 2.36. Notably, the model achieved the best performance when both average pooling and identity mapping

were applied simultaneously, yielding an EER of 2.24. These results indicate that the inclusion of identity mapping and average pooling operators enhances the model's ability to extract speaker feature representations.In this study, we also attempted to replace all average pooling operators in the self-calibrated convolution with the maximum pooling operator to observe potential performance differences. As depicted in Table 5, the introduction of the maximum pooling operator, alongside other configurations, resulted in a decrease in EER to 2.33. This article argues that, unlike maximum pooling, average pooling establishes connections between locations within the entire pooling window, allowing for a better capture of local contextual information.

5. Conclusions

This paper introduces a novel self-calibrating convolutional module with PConv, which adaptively constructs long-range time-frequency and channel dependencies around each time-frequency location. Building on this module, we propose the SpcNet model, designed to capture comprehensive feature representations by expanding the receptive field of the network. The SpcNet model effectively leverages the nested convolutional filters within the convolutional layers, extracting multi-scale feature representations to enhance model performance. The experimental results verify that the method proposed in this paper is effective.

References

Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S-H Gary Chan. Run, don't walk: chasing higher flops for faster neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12021–12031. IEEE, Los Alamitos, CA, 2023.

Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. In *IEEE Transactions on Audio, Speech, and Language Processing*, volume 19, pages 788–798. IEEE, Piscataway, NJ, 2010.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. In *Interspeech 2020*, pages 3830–3834. ISCA-INT Speech Communication Assoc, France, 2020.

George R Doddington, Mark A Przybocki, Alvin F Martin, and Douglas A Reynolds. The nist speaker recognition evaluation – overview, methodology, systems, results, perspective. In *ESCA Workshop on Speaker Recognition and its Commercial and Forensic Applications (RLA2C)*, volume 31, pages 225–254. ACM, 2000.

Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. In *IEEE transactions on pattern analysis and machine intelligence*, volume 43, pages 652–662. IEEE, Los Alamitos, CA, 2019.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (2017 ICASSP)*, pages 5220–5224. Piscataway, NJ, 2017.

Bei Liu, Zhengyang Chen, Shuai Wang, Haoyu Wang, Bing Han, and Yanmin Qian. Df-resnet: Boosting speaker verification performance with depth-first design. In *Interspeech*, pages 296–300. ISCA-INT Speech Communication Assoc, France, 2022a.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986. IEEE, Los Alamitos, CA, 2022b.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019*. ISCA-INT Speech Communication Assoc, France, 2019.

Douglas A Reynolds and Richard C Rose. Robust text-independent speaker identification using gaussian mixture speaker models. In *IEEE transactions on speech and audio processing*, volume 3, pages 72–83. IEEE, Piscataway, NJ, 1995.

Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. In *5th Annual NIST Speaker Recognition Workshop*, volume 10, pages 19–41. Academic Press inc Elsevier Science, 2000.

David Snyder, Guoguo Chen, and Daniel Povey. Musan: A music, speech, and noise corpus. *Computer Science, Chongqing. arXiv preprint arXiv:1510.08484*, 2015.