STATISTICAL PYRAMID DENSE TIME DELAY NEURAL NETWORK FOR SPEAKER VERIFICATION

Zi-Kai Wan[1], *Qing-Hua Ren*[1], *You-Cai Qin*[1], *Qi-Rong Mao*[1,2,*]

[1]School of Computer Science and Communication Engineering, Jiangsu University, China
[2]Jiangsu Engineering Research Center of Big Data
Ubiquitous Perception and Intelligent Agriculture Applications, China
[*]Corresponding Author: mao_qr@ujs.edu.cn

ABSTRACT

Recently, speaker verification (SV) techniques relay on deep learning frameworks to extract more informative embedding vectors, which greatly improves the accuracy compared with traditional machine learning methods. The well-known x-vector architecture, a time delay neural network (TDNN), is widely adapted for SV tasks. However, most of existing variants rarely combines the global and sub-region context information and suffer from the local receptive field that is engendered by the standard convolutional operation. In this paper, we propose statistical pyramid dense TDNN (SPD-TDNN) with the statistical pyramid pooling module which captures the context information. Specifically, the developed module adaptively exchanges information among contextual regions from different perspectives, which correspond to multiple parallel branches. The statistics collected by the global-region branch are comprised of mean and standard deviation across the time domain to acquire the more global context information. Extensive experiments on the VoxCeleb1&2 datasets demonstrate that the proposed PSD-TDNN outperforms corresponding D-TDNN, D-TDNN-SS and ECAPA-TDNN which achieve the state-of-the-art performances on the SV task, with similar model complexity.

Index Terms— speaker verification, global context information, statistical pyramid pooling block, SPD-TDNN

1. INTRODUCTION

Speaker verification(SV) aims to automatically judge whether a pair of utterances belong to the same speaker or not, usually by scoring the similarity between enrollment and test utterances. The task of SV is usually divided into two steps. The first step is to extract fixed-length speaker embedding vectors from variable-length utterances. The deep neural networks [1, 2, 3, 4, 5, 6] serve for mainstream extractions. The second step is to calculate the similarity between two embedding vectors. Representative scoring methods include cosine similarity and probabilistic linear discriminant analysis (PLDA) [7].

Most of the existing advanced SV frameworks benefit from the convolutional neural network (CNN), yielding the state-of-the-art performance. These methods boost dynamic object understanding, and yet still face a vexed problem. It is shown by Zhou *et al.* [8] that empirical receptive field of CNN is much smaller than the theoretical one especially on high-level layers. The standard convolutional operation may suffer from the local receptive field, leading to the lack of the momentous global contextual prior. However, most existing methods follow such a paradigm and produce suboptimal results.

In recent studies, the researchers have explored to weave the global context information into time delay neural network (TDNN)-based models. For example, the ECAPA-TDNN [9] fuses the Squeeze-and-Excition [10] block (SE block) to explicitly model channel interdependencies. The squeeze operation simply calculates the mean vector of the frame-level features across the time domain. The densely connected TDNN-statistics-and-selection (D-TDNN-SS) [11, 12] combines the high-order statistics pooling (HOSP) layer [11, 12] which collects mean, standard deviation, skewness and kurtosis information for each channel to integrate the global information.

The models mentioned above achieve the state-of-the-art results in addressing the task of SV. However, the method adopted to extract the context information is subject to further improvement. In mainstream literature, the extra layer often follows some CNN layers, which is used to extract the global context information. However, engendered by defects of the CNN layers, the lack of empirical receptive field size, the information still has been lost to a certain degree.

To address this limitation, alternatives are explored by changing the series framework between the global layer and the CNN layers. Inspired by the success of pyramid pooling block [13], we make the following contributions: (1) Different from the mainstream methods, we propose the statistical pyramid dense TDNN (SPD-TDNN) with statistical pyramid pooling module. The complete context features is extracted by the average pooling layers, and then concatenated with the input features prior to being fed into the CNN layer. (2) To incorporate suitable global features, the developed module adaptively exchanges information among contextual regions from global and sub-region perspectives, which correspond to multiple parallel branches. (3) To complement the global context information, standard deviation is added into the branch with the global region, which describes the frame-level statistical dispersion across the time domain.

The effectiveness of the proposed method is then evaluated by experiments on the VoxCeleb1 [14] and VoxCeleb2 [15] datasets. Further improvements on SV performance of the proposed SPD-TDNN network are confirmed by the experimental results, as presented in the end of this paper.

2. METHODS

In this section, the paper provides a specific structural description of the proposed SPD-TDNN with three subsections. The first one describes the constituent parts and the detailed structure of the SPD-TDNN. The second makes the detailed introduction to the statistical pyramid pooling block. The last one introduces the Angular additive margin softmax (AAM-Softmax) [16] which is used to classify the speakers.

Table 1. The structure of SPD-TDNN. '$Input'$ denotes the input channels size. '$Output'$ means the output channels size. C, k, p, K respectively represent the number of channels, kernel size, padding size and the number of classes. '$Parameters'$ of SPD-TDNN layers is shown as ($input$, $growth\ rate$, $dilation$).

#	Style Name	Input	Output	Parameters
1	Conv1d+BN+RELU	C	128	k=5, p=2
2	SPD-TDNN [Fig. 1]	128	192	(128, 64, 1)
	SPD-TDNN	192	256	(192, 64, 2)
	SPD-TDNN	256	320	(256, 64, 3)
	SPD-TDNN	320	384	(320, 64, 1)
	SPD-TDNN	384	448	(384, 64, 2)
	SPD-TDNN	448	512	(448, 64, 3)
	Conv1d+BN+RELU	512	256	k=1, p=0
	SPD-TDNN	256	320	(256, 64, 1)
	SPD-TDNN	320	384	(320, 64, 2)
	SPD-TDNN	384	448	(384, 64, 3)
	SPD-TDNN	448	512	(448, 64, 1)
	SPD-TDNN	512	576	(512, 64, 2)
	SPD-TDNN	576	640	(576, 64, 3)
	SPD-TDNN	640	704	(640, 64, 1)
	SPD-TDNN	704	768	(704, 64, 2)
	SPD-TDNN	768	832	(768, 64, 3)
	SPD-TDNN	832	896	(832, 64, 1)
	SPD-TDNN	896	960	(896, 64, 2)
	SPD-TDNN	960	1024	(960, 64, 3)
	Conv1d+BN+RELU	1024	512	k=1, p=0
3	Statistic Pooling + BN	512	1024	
4	FC + BN	1024	128	
	AAM-Softmax		K	

2.1. Overview of SPD-TDNN

The architecture of SPD-TDNN is presented in Table 1, which consists of four main components. The technical details are shown as follows:

(1) In the first one, the CNN layer is used to initialize the number of channels to fixed-size dimension. The kernel size is set as 5, the padding is set as 2, and the dilation is set as 1.

(2) In the second component, every three SPD-TDNN layers are constituted into a group, and the sizes of dilation are set as 1, 2, and 3 respectively. The different sizes of dilation remaining the parameters and the calculation efficiency unchanged make the model archieve different receptive fields. These groups combined with CNN layers constitute two SPD-TDNN blocks in total. The first block is constructed with two SPD-TDNN groups and one CNN layer; the second contains four groups followed by one CNN layer.

(3) The third component is the proposed statistics pooling layer, which aggregates frame-level features and outputs utterance-level features.

(4) The last one is a fully connected (FC) layer, which is used to output fixed-length speaker embedding vectors.

Specifically, SPD-TDNN layer is the basic unit of SPD-TDNN model. Each SPD-TDNN layer consists of a consecutive feed-forward neural network (FNN)-based bottleneck layer and a statistical pyramid pooling block. The components of SPD-TDNN layer is shown in Fig. 1. The output size of each statistical pyramid pooling block is called growth rate which is supposed as g. The output size of each bottleneck layer is set as be twice of the growth rate, i.e., $2g$. Finally, the input of SPD-TDNN layer and the output of statistical pyramid pooling block will be concatenated.

Multiple SPD-TDNN layers followed by a CNN layer whose kernel size is set as 1 construct a SPD-TDNN block. The CNN layer can be used to aggregate multi-stage features from different layers. The output of l-th SPD-TDNN layer is shown as follow:

$$p^l = P^l([p^0, p^1, ..., p^{l-1}]) \tag{1}$$

Where p^l denotes the output of l-th SPD-TDNN layer. $P^l(\cdot)$ denotes the non-linear transform of l-th layer. $[\cdot]$ denotes the concatenation operation.

2.2. Statistical pyramid pooling block

This subsection provides the specific introduction of the statistical pyramid pooling block. The block obtains the global and sub-region context information from the sentences to capture more comprehensive feature representations, which is helpful to distinguish the categories of different speakers. The structure of the improved block is shown in Fig. 1.

The input feature is first fed into multiply parallel branches for the building of the relationship between frames to collect the global and sub-region context information. The output

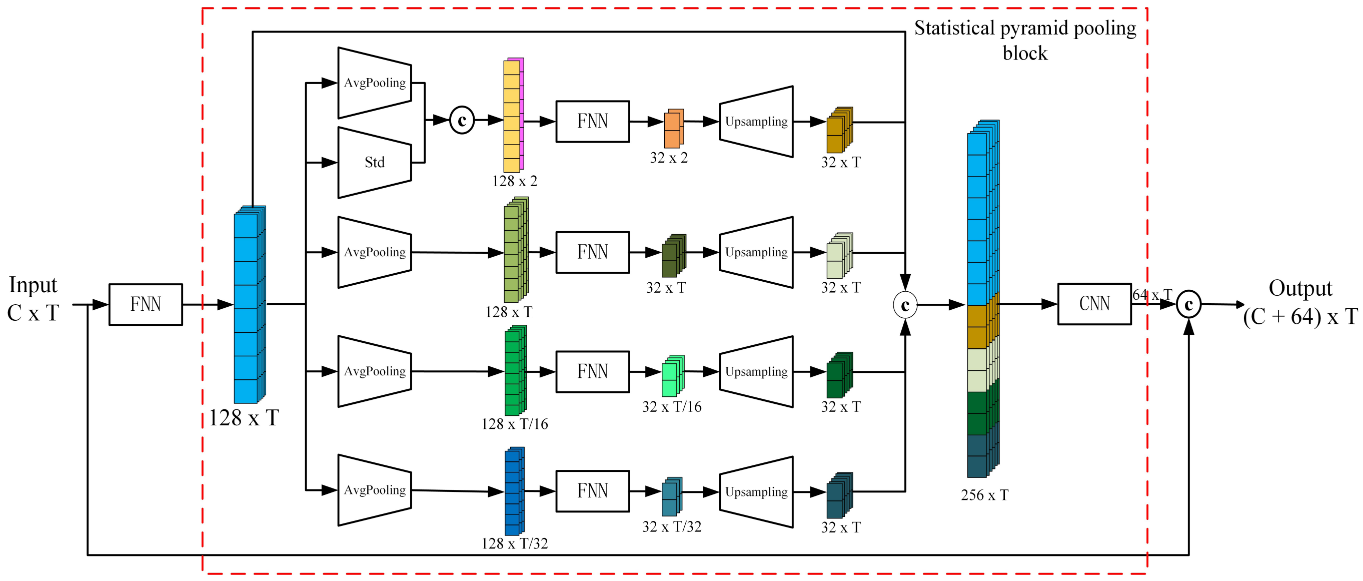

Fig. 1. The structure of the SPD-TDNN layer. FNN is the bottleneck layer, and the pyramid pooling block is in the red dotted box. C and T represent the number of channels and frame length of the input feature. $'©'$ means concatenation operation.

of parallel branches are concatenated as the context features of the sentence. Finally, the context features is concatenated with the input feature map and fed into the CNN layer whose kernel size is set as 3 for the fusion and extraction of features.

Specifically, the parallel branches include one global-region branch and multiply sub-region branches. The average pooling layer of the global-region branch compresses the frames across the time domain into the mean. The mean is an indicator that reflects the central tendency of frame-level features. In order to supplement more global context information, the standard deviation is added as a measure of statistical dispersion to reflect the degree of dispersion between individuals in the feature. These statistics are concatenated together as the global vector. The pooling layers of sub-region branches separate the feature map into different regions and form pooled representation for different locations. In order to maintain the weight of global features, the FNN layer is added after the pooling layers to change dimension of context representation into $\frac{1}{N}$ of the original one if the number of branches is N. Then, the low-dimension features are upsampled with the same frame length as the original feature via bilinear interpolation to be the output of parallel branches.

Noted that the number of statistical pyramid branches and size of each branch can be modified. The structure abstracts different sub-regions by adopting varying-size pooling kernels in a few strides. Therefore, the gap of kernel size should be maintained to ensure the diversity and complementarity of features. Our statistical pyramid pooling layer contains four parallel branches. Excluding the global-region branch, the kernel sizes of average pooling layers used in sub-region branches are set as 1, 16, and 32 respectively.

2.3. AAM-Softmax

Softmax-loss function is commonly employed in the classification of speakers. The fixed-length output embedding of the SPD-TDNN is fed into the loss function. AAM-Softmax is one of the kinds of Softmax loss based on cosine similarity calculation [16, 17]:

$$L_{AAM\_Softmax} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{(\cos(\theta_{y_i,i}+m))}}{Z} \tag{2}$$

Where $Z = e^{\cos(\theta_{y_i,i}+m)} + \sum_{j=1,j\neq i}^{c} e^{s}(\cos(\theta_{j,i}))$. N is the general category, s is a scaling factor, which is used to ensure that the gradient in training is not too small, and m is a margin between different classes. $\theta_{j,i}$ represents the Angle between the weight vector W_j and the feature vector X_i.

3. EXPERIENTAL SETUP

3.1. Dataset and input features

The performance of the SPD-TDNN is evaluated on VoxCeleb1 and VoxCeleb2 datasets. Each dataset contains a development set and a test set. We use Kaldi [18] to generate the development set. Specifically, the two datasets are combined to form a larger training set. This development set contains more than 1,200,000 utterances from 7,323 speakers. 0.4% of the utterances are divided as the valid set. The test set uses the

Table 2. Results on the VoxCeleb1 test set.

Model	Parameters(M)	EER(%)	MinDCF(0.01)	MinDCF(0.001)
D-TDNN [11, 12]	2.43	1.65	0.18	0.27
D-TDNN-SS [11, 12]	3.10	1.27	0.14	**0.20**
ECAPA-TDNN [9]	5.86	1.52	0.16	0.22
SPD-TDNN(no std)	3.16	1.30	0.15	0.21
SPD-TDNN	3.16	**1.20**	**0.13**	0.22

test set of VoxCeleb1, which includes 37,720 test pairs from 40 speakers.

The 30-dimensional Mel-frequency cepstrum coefficients (MFCCs) over a 25ms long window every 10ms are calculated by Kaldi. Cepstral mean normalization (CMN) is used in a 3-second sliding window, and energy-based voice activity detection (VAD) [19] is used to remove silent frames.The length of the spectrograms is randomly split between 200 and 400 frames.

3.2. Model configuration

We implement the D-TDNN [11, 12], D-TDNN-SS [11, 12] and ECAPA-TDNN [9] models in Pytorch. The D-TDNN is used as a baseline. To compare with the baseline, the parameters are set according to the experiment in [11].

The stochastic gradient descent (SGD) optimization model is used with momentum set as 0.95 and weight decay set as 5e-4. Mini batch is set as 128 and the learning rate is initialized to 0.01 and divided by 10 at 120K-th and 180K-th iterations. The training ended at the 240K-th iteration. The size of the speaker embedding vector output by the final model is set as 128.

Every model uses AAM-Softmax loss function to classify speakers. And for the AAM-Softmax loss, the margin and scaling parameters are set as 0.2 and 30, respectively.

3.3. Model testing and evaluation criteria

In the training process, the accuracy calculated by the valid set is used as the evaluation standard for selecting the model parameters. The performance is evaluated by equal error rate (EER) [20] and minimum detection cost function (MinDCF) [20], which are calculated by the consine method.

3.4. Performance and evaluation

The results are shown in Table 2. We find that the proposed SPD-TDNN model is superior to other test methods in almost all indicators.

Specifically, compared with the baseline D-TDNN model, the parameters of our model only increased by 0.73M, and the EER decreased by 0.45% to 1.20%. This shows that the statistical pyramid pooling block plays a great role in the SPD-TDNN.

As a variant of D-TDNN model, D-TDNN-SS model has the parameters of 3.10M and the EER of 1.27%. Compared with the model, the EER of SPD-TDNN is decreased by 0.7% when the parameters are only increased by 0.06M. The results obtained by SPD-TDNN model for MinDCF (0.001) and MinDCF (0.01) do not differ much from those obtained by D-TDNN-SS, which are in a reasonable fluctuation range. On the other hand, compared with D-TDNN and ECAPA-TDNN, it can be seen that in general the EER and MinDCF are in a normal decreasing trend. Our SPD-TDNN can outperform the TDNN-based models which have the state-of-the-art accuracy. The experiments verify the effectiveness of the context information extracted by the proposed model.

The result of SPD-TDNN without standard deviation is worse than the proposed SPD-TDNN. The comparative experiment confirms the importance of standard deviation for the global context information.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we propose the SPD-TDNN with the statistical pyramid pooling block for the SV task. The improved block provides more superior global context information. On the other hand, the multi-branch structure of the statistical pyramid module also enables the model to obtain sub-region information containing different region sizes. Through experiments, we confirm that by combining the statistical pyramid pooling module with D-TDNN, SPD-TDNN can more fully integrate the global and sub-region context information, and can obtain more advanced results with small amount of parameters.

In the future, we hope to explore more methods that can integrate global context information to better solve the problem of CNN's real perception field.

5. ACKNOWLEDGEMENTS

This work is supported in part by the Key Projects of the National Natural Science Foundation of China under Grant U1836220, the National Nature Science Foundation of China under Grant 62176106. Jiangsu Province key research and development plan under Grant BE2020036.

6. REFERENCES

[1] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[2] David Snyder, Pegah Ghahremani, Daniel Povey, Daniel Garcia-Romero, Yishay Carmiel, and Sanjeev Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 165–170.

[3] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[4] Ya-Qi Yu, Lei Fan, and Wu-Jun Li, "Ensemble additive margin softmax for speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6046–6050.

[5] Zhuo Chen, Takuya Yoshioka, Liang Lu, Tianyan Zhou, Zhong Meng, Yi Luo, Jian Wu, Xiong Xiao, and Jinyu Li, "Continuous speech separation: Dataset and analysis," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7284–7288.

[6] Ying Liu, Yan Song, Ian McLoughlin, Lin Liu, and Li-rong Dai, "An effective deep embedding learning method based on dense-residual networks for speaker verification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6683–6687.

[7] Simon JD Prince and James H Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[8] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, "Object detectors emerge in deep scene cnns," *arXiv preprint arXiv:1412.6856*, 2014.

[9] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *INTERSPEECH*, 2020, pp. 1–5.

[10] J Hu, L Shen, S Albanie, G Sun, and E Wu, "Squeeze-and-excitation networks.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2019.

[11] Ya-Qi Yu and Wu-Jun Li, "Densely connected time delay neural network for speaker verification.," in *INTERSPEECH*, 2020, pp. 921–925.

[12] Tianlong Kong, Shouyi Yin, Dawei Zhang, Wang Geng, Xin Wang, Dandan Song, Jinwen Huang, Huiyu Shi, and Xiaorui Wang, "Dynamic multi-scale convolution for dialect identification," *arXiv preprint arXiv:2108.07787*, 2021.

[13] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[14] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[15] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[16] Xu Xiang, Shuai Wang, Houjun Huang, Yanmin Qian, and Kai Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 1652–1656.

[17] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[18] Mirco Ravanelli, Titouan Parcollet, and Yoshua Bengio, "The pytorch-kaldi speech recognition toolkit," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6465–6469.

[19] Shaojin Ding, Quan Wang, Shuo-yiin Chang, Li Wan, and Ignacio Lopez Moreno, "Personal vad: Speaker-conditioned voice activity detection," *arXiv preprint arXiv:1908.04284*, 2019.

[20] David Snyder, Daniel Garcia-Romero, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5796–5800.