INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Improving Deep CNN Architectures with Variable-Length Training Samples for Text-Independent Speaker Verification

Yanfeng Wu[1,†] , *Junan Zhao*[1,†] , *Chenkai Guo*[2,*] , *Jing Xu*[1,*]

[1]College of Artificial Intelligence, Nankai University, Tianjin, China
[2]College of Computer Science, Nankai University, Tianjin, China
{yanfeng_wu,zja}@mail.nankai.edu.cn, {guochenkai,xujing}@nankai.edu.cn

Abstract

Deep Convolutional Neural Network (CNN) based speaker embeddings, such as r-vectors, have shown great success in text-independent speaker verification (TI-SV) task. However, previous deep CNN models usually use fixed-length samples for training and employ variable-length utterances for speaker embeddings, which generates a mismatch between training and embedding. To address this issue, we investigate the effect of employing variable-length training samples on CNN-based TI-SV systems and explore two approaches to improve the performance of deep CNN architectures on TI-SV through capturing variable-term contexts. Firstly, we present an improved selective kernel convolution which allows the networks to adaptively switch between short-term and long-term contexts based on variable-length utterances. Secondly, we propose a multi-scale statistics pooling method to aggregate multiple time-scale features from different layers of the networks. We build a novel ResNet34 based architecture with two proposed approaches. Experiments are conducted on the VoxCeleb datasets. The results demonstrate that the effect of using variable-length samples is diverse in different networks and the architecture with two proposed approaches achieves significant improvement over r-vectors baseline system.

Index Terms: speaker verification, CNN, selective kernel, multi-scale aggregation, statistics pooling

1. Introduction

Speaker verification (SV) is a task of verifying a person's claimed identity from speech signals. According to different application scenarios, SV can be generally classified into two typical categories: text-dependent SV (TD-SV) and text-independent SV (TI-SV). TD-SV restrains the texts of utterances from being fixed for enrollment and verification; while TI-SV has no such restriction on the texts of utterances, which is the focus of this work. Over the past decade, the dominant method for TI-SV is the combination of i-vector-based representation [1] and a probabilistic linear discriminant analysis (PLDA) [2] backend classifier. Recently, the most widely used approach for TI-SV is the deep speaker embedding which adopts deep neural network (DNN) instead of i-vector to extract speaker representations from utterances and applies the PLDA or cosine distance for similarity scoring. This method outperforms i-vector-based method, especially when a large number of training utterances are available.

Currently, the two state-of-the-art DNN-based methods for TI-SV are x-vectors [3] based on time-delay neural network (TDNN) and r-vectors [4] based on deep convolutional neural network (CNN). The x-vector employs five TDNN layers to extract frame-level features, a statistics pooling layer to aggregate frame-level outputs across all frames into a fixed-dimensional utterance-level vector, and two fully connected (FC) layers to extract speaker embeddings. Different from x-vector, the r-vector accepts three-dimensional features as input and adopts a residual network with 34 parameter layers (ResNet34) [5]. R-vector based TI-SV system achieves the best performance in 2019 VoxCeleb Speaker Recognition Challenge (VoxSRC 2019) [6], showing great success of deep CNN architectures in handling the TI-SV task.

During the training of DNN-based TI-SV systems, the utterances in the training dataset are cut into segments as training samples. For TDNN-based models, the length of training samples is usually variable from 2s to 4s. In contrast, most previous CNN models [4, 7, 8, 9, 10] use fixed-length samples for training, and apply variable-length utterances for speaker embeddings, which generates a mismatch between training and embedding. In addition, fixed-length training samples make it difficult for the networks to model variable-term contexts, reducing the performance on TI-SV task. This problem has not received sufficient attention in prior works about CNN-based TI-SV systems.

In this paper, based on above observations, we investigate the effect of employing variable-length training samples on CNN-based TI-SV systems and explore two novel approaches to improve the performance of deep CNN architectures on TI-SV via capturing variable-term contexts. Firstly, we present an improved selective kernel convolution based on Selective Kernel Networks [11], which allows the networks to adaptively switch between short-term and long-term contexts based on variable-length utterances. Secondly, we propose a multi-scale statistics pooling method to aggregate multiple time-scale features from different layers of the networks. We build a novel ResNet34 based architecture with two proposed approaches. Experiments are conducted on public VoxCeleb datasets [12, 13, 6]. The results show that the effect of using variable-length samples is diverse in different networks and the architecture with two proposed approaches achieves significant improvement over r-vectors baseline system.

The remainder of this paper is organized as follows. Section 2 describes the network with two proposed methods in detail. Sections 3 introduces the experimental setup. The results and analysis are presented in Section 4. Finally, the conclusions are given in Section 5.

2. Approaches

In this section, we describe two proposed approaches: improved selective kernel convolution and multi-scale statistics pooling. Inspired by r-vectors [4], we construct a ResNet34 architecture

[†] equal contribution
[*] corresponding authors

Copyright © 2021 ISCA

http://dx.doi.org/10.21437/Interspeech.2021-559

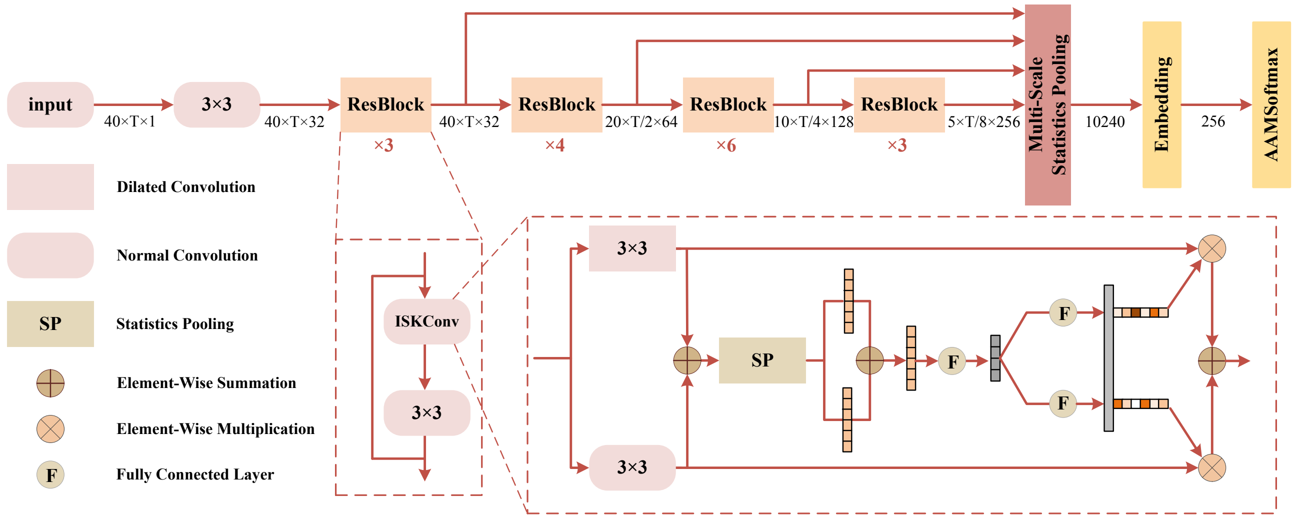

Figure 1: *Illustration of the ResNet34 architecture with the two proposed approaches.*

with the two proposed methods, as illustrated in Figure 1.

2.1. Improved selective kernel convolution

Dilated convolution is widely employed in CNN-based models for TI-SV [14, 15, 16], where the filter is applied over an area larger than its length by skipping input values with a certain step. Dilated convolution can acquire a larger receptive field size than normal convolution, which is beneficial for the network to capture long-term contexts for variable-length utterances. However, exploiting too many dilated convolutional layers may lead to the "gridding" problem [17] and lose some important neighboring information, especially in higher layers of deep CNN architectures.

To address this problem, we introduce an improved selective kernel convolution (ISKConv) based on [11], which performs normal convolution and dilated convolution in two paths simultaneously and applies a dynamic channel-wise selection mechanism based on softmax attention. The improved selective kernel convolution can make the networks adaptively switch between short-term and long-term contexts based on variable-length input utterances.

Suppose the input feature of ISKConv is $\mathbf{X} \in \mathbb{R}^{T' \times F' \times C'}$, where T',F',C' denote the dimension along time, frequency and channel axes, respectively. We first conduct a 3×3 convolution $\widehat{F}$ and a 3×3 convolution $\widetilde{F}$ with a dilation rate of 2:

$$\delta(B(\widehat{F})) : \mathbf{X} \to \widehat{\mathbf{U}} \in \mathbb{R}^{T \times F \times C}, \tag{1}$$

$$\delta(B(\widetilde{F})) : \mathbf{X} \to \widetilde{\mathbf{U}} \in \mathbb{R}^{T \times F \times C}. \tag{2}$$

In Equation 1 and 2, B and δ denote batch normalization and ReLU function, respectively. We exploit 3×3 dilated convolutions rather than 5×5 normal convolutions in SKNet [11] because two kinds of convolutions have the same reception field size and dilated convolutions have less parameters. Then we employ an element-wise summation to integrate information from two branches:

$$\mathbf{U} = \widehat{\mathbf{U}} \oplus \widetilde{\mathbf{U}}, \tag{3}$$

where $\oplus$ denotes the element-wise summation. In Equation 3, $\mathbf{U}$ is the global representation obtained from two paths.

Next we apply a soft attention across the channel axis based on the global feature $\mathbf{U}$. Different from [11] which uses global average pooling (GAP) to generate channel-wise statistics $\mathbf{s} \in \mathbb{R}^C$, we perform statistics pooling (SP) [3] that has been proved to be more effective in TI-SV task. Suppose that c is the channel index, we first apply average pooling across the frequency axis:

$$\mathbf{h}_c(t) = \frac{1}{F}\sum_{f=1}^{F} \mathbf{U}_c(t, f), \tag{4}$$

where $\mathbf{h}_c(t)$ is the feature vector at the c-th channel along the time axis. Then, the mean and standard deviation of the feature vector is obtained:

$$m_c = \frac{1}{T}\sum_{t=1}^{T} \mathbf{h}_c(t), \tag{5}$$

$$d_c = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \mathbf{h}_c(t) \cdot \mathbf{h}_c(t) - m_c \cdot m_c}. \tag{6}$$

Next, we fuse the outputs of the pooling operation. In this paper, we simply utilize a element-wise summation which does not increase the parameters of the network compared with global average pooling:

$$s_c = m_c + d_c. \tag{7}$$

The s_c denotes c−th element of the vector $\mathbf{s}$.

Further, we perform a multi-layer perceptron (MLP) which contains two FC layers. The first FC layer applies channel-downscaling with reduction ratio r:

$$\mathbf{z} = \delta(B(\mathbf{W}\mathbf{s})), \tag{8}$$

where $\mathbf{W} \in \mathbb{R}^{d \times C}$ is the weight matrix, and d is calculated as follows:

$$d = \max(\frac{C}{r}, L), \tag{9}$$

where L is a hyperparameter. In our experiments, r is set to 16 and L is set to 32.

Finally, we perform an FC layer for each branch respectively and execute a softmax function to ensure that the sum of all weights on each channel is equal to 1:

$$a_c = \frac{e^{\mathbf{A}_c\mathbf{z}}}{e^{\mathbf{A}_c\mathbf{z}} + e^{\mathbf{B}_c\mathbf{z}}}, b_c = \frac{e^{\mathbf{B}_c\mathbf{z}}}{e^{\mathbf{A}_c\mathbf{z}} + e^{\mathbf{B}_c\mathbf{z}}}, a_c + b_c = 1 \tag{10}$$

where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{C \times d}$ denote the soft attention matrices for $\widehat{U}$ and $\widetilde{U}$, respectively. $\mathbf{A}_c, \mathbf{B}_c \in \mathbb{R}^{1 \times d}$ is the c-th row of $\mathbf{A}$ and $\mathbf{B}$, respectively.

The final output $\mathbf{V}$ of ISKConv is obtained as follows:

$$\mathbf{V}_c = a_c \cdot \widehat{\mathbf{U}}_c \oplus b_c \cdot \widetilde{\mathbf{U}}_c, \tag{11}$$

where $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_C]$ and $\mathbf{V}_c \in \mathbb{R}^{T \times F}$.

The ISKConv is easily integrated into the residual blocks. In our implementation, we replace the first convolution layer of each residual block in ResNet34 with ISKConv, as shown in Figure 1.

2.2. Multi-scale statistics pooling

Multi-scale aggregation (MSA) [18, 19, 20, 10] is another introduced method for CNN-based networks, which utilizes multi-scale features from different layers of CNN. In r-vectors [4], a statistics pooling method is proposed for CNN-based features and shows superior performance. Since statistics pooling can capture the speaker's characteristics in terms of the temporal variations over long-term contexts, we design its multi-scale version to collect various temporal information from multiple time-scale features.

We denote the output of each residual module as $\mathbf{X}_i \in \mathbb{R}^{T_i \times F_i \times C_i}$ for $i = 1, 2, 3, 4$, where T_i, F_i, C_i denote the dimension of time, frequency and channel axes, respectively. We first apply statistics pooling along time axis to obtain a mean feature $\boldsymbol{\mu}_i \in \mathbb{R}^{1 \times F_i \times C_i}$ and a standard deviation feature $\boldsymbol{\sigma}_i \in \mathbb{R}^{1 \times F_i \times C_i}$ for each output:

$$\boldsymbol{\mu}_i(f, c) = \frac{1}{T} \sum_{t=1}^{T} \mathbf{X}_i(t, f, c), \forall i \in \{1, 2, 3, 4\}, \tag{12}$$

$$\boldsymbol{\sigma}_i(f, c) = \sqrt{\frac{1}{T} \sum_{t=1}^{T} (\mathbf{X}_i(t, f, c) - \boldsymbol{\mu}_i(f, c))^2}, \forall i \in \{1, 2, 3, 4\}, \tag{13}$$

where t, f, c represent the index of time, frequency and channel, respectively. Then we concatenate the two features along the time axis and use a flatten operation to transform $\mathbf{P}_i$ into a vector:

$$\mathbf{P}_i \in \mathbb{R}^{2 \times F_i \times C_i} \rightarrow \mathbf{E}_i \in \mathbb{R}^{d_i}, \forall i \in \{1, 2, 3, 4\} \tag{14}$$

where $d_i = 2 \cdot F_i \cdot C_i$.

The final output $\mathbf{E}$ of multi-scale statistics pooling is the concatenation of all output vectors:

$$\mathbf{E} = [\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3, \mathbf{E}_4]. \tag{15}$$

3. Experimental setup

3.1. Datasets

The experiments are conducted on public VoxCeleb1 [12] and VoxCeleb2 dataset [13]. Each dataset contains a development set and a test set. We select the development set of VoxCeleb2 as the training set, which includes 5994 speakers and over one million utterances. We do not employ any techniques to augment the training data due to the limitation of computing resources. The test set is the whole VoxCeleb1 dataset. We select three trial lists provided by VoxSRC 2019 challenge [6] to verify the performance of different models on TI-SV task: cleaned VoxCeleb1 test set (VoxCeleb1-O), cleaned extended VoxCeleb1 test set (VoxCeleb1-E) and cleaned hard VoxCeleb1 test set (VoxCeleb1-H). The detailed information of the training and test set is shown in Table 1.

Table 1: *Dataset for training and evaluation.*

Dataset	Speakers	Utterances	Trials
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

The feature extraction step is handled with Kaldi toolkit [21]. The input acoustic features are 40-dimensional Mel-filterbanks (FBanks) from 25ms windows with 10ms shift between frames. In addition, we exploit cepstral mean normalization (CMN) with a 3-second window, and adopt energy-based voice active detection (VAD) to remove the silent frames.

3.2. Implementation details

The network training and embedding extraction of all systems are implemented by an existing Tensorflow tookit[1] [22]. All models are trained with the stochastic gradient descent (SGD) optimizer and the batch size is set to 64. The learning rate is initialized as 0.01, and then is continuously reduced once the validation loss fails to decrease for a while, till it goes down below 1e-6. We perform a weight decay of 1e-2 in all models to reduce overfitting. In order to obtain more discriminative speaker embeddings, we exploit Additive Angular Margin Softmax (AAM-Softmax) loss [22, 23]:

$$L = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{s(\cos(\theta_{y_i} + m))}}{e^{s(\cos(\theta_{y_i} + m))} + \sum_{j \neq y_i} e^{s \cos \theta_j}} \tag{16}$$

where s is a scale factor and m is the margin. In our experiments, the margin is set to 0.2 and the scale factor is the feature norm of the last output layer.

We apply cosine distance as the backend to compute verification scores for all comparative systems. The extracted embeddings are centered with the mean of all training utterances. In addition, we adopt adaptive symmetric score normalization [24] for better performance. The cohort is created by averaging the length-normalized embeddings of 2k randomly-selected training utterances and the size of the top cohort is set to 100.

We report the experimental results in terms of the Equal Error Rate (EER) and the minimum of normalized detection cost function (MinDCF) with $P_{target} = 0.01, C_{FA} = C_{Miss} = 1$.

4. Results

We exploit two kinds of training samples, fixed-length and variable-length, to train each model respectively. The duration of fixed-length training samples is 2s corresponding to 200 frames, while the length of variable-length training samples varies from 2s to 4s, corresponding to 200 frames to 400 frames. Note that the temporal dimension of input features is fixed in each batch, but varies in different batches. We evaluate four different architectures: ResNet34 [4], ResNet34 with improved selective kernel convolutions (ResNet34+ISKConv), ResNet34 with multi-scale statistics pooling (ResNet34+MSSA) and

[1] https://github.com/entn-at/tf-kaldi-speaker

Table 2: *Experimental results on VoxCeleb1 with different architectures. ISKConv, MSSP, Training Length denote improved selective kernel convolution, multi-scale statistics pooling and the frame length of training samples, respectively.*

Model	**Param**	**Training Length**	**VoxCeleb1-O**		**VoxCeleb1-E**		**VoxCeleb1-H**	
			EER%	**MinDCF**	**EER(%)**	**MinDCF**	**EER(%)**	**MinDCF**
ResNet34 [4]	6.0M	200	1.526	0.1736	1.580	0.1828	2.771	0.2630
		200-400	1.558	0.1821	1.578	0.1735	2.681	0.2499
ResNet34 +ISKConv	8.2M	200	1.521	0.2160	1.590	0.1760	2.776	0.2559
		200-400	1.372	**0.1509**	1.480	0.1655	2.531	0.2423
ResNet34 +MSSP	7.9M	200	1.473	0.1871	1.520	0.1707	2.719	0.2574
		200-400	1.521	0.2022	1.560	0.1656	2.737	0.2477
ResNet34 +ISKConv+MSSP	10.1M	200	**1.266**	0.1673	1.320	0.1551	2.471	0.2317
		200-400	1.292	0.1618	**1.319**	**0.1463**	**2.396**	**0.2219**

ResNet34 with both methods (ResNet34+ISKConv+MSSP). The results are listed in Table 2.

We firstly investigate the effect of using variable-length training samples on different networks. Applying variable-length samples as training data can improve the performance of ResNet34+ISKConv significantly, but can not obtain significant improvement on other three networks. The result indicates that the influence of variable-length training technique is diverse in different architectures. Moreover, we observe that variable-length training can contribute to performance improvement on VoxCeleb1-H test sets.

Next, we mainly focus on the proposed network ResNet34+ISKConv+MSSP, which obtains the best performance in almost all metrics except MinDCF in VoxCeleb1-O. Specifically, the proposed network trained with variable-length training samples achieves EER of 1.292%, 1.319% and 2.396% in three test sets, which corresponds to 15.3%, 16.5% and 10.6% reduction over the ResNet34 model. The proposed network also achieves 6.8%, 15.7% and 11.2% relative improvement in MinDCF over ResNet34, demonstrating the effectiveness of two proposed methods.

We conduct an ablation study of the proposed approaches in order to evaluate the performance of ISKConv and MSSP respectively. Next we compare all networks under the condition of employing variable-length training samples. When MSSP is removed from the proposed network, the EER is increased by 6.2%, 12.2% and 5.6% in three test sets, respectively. This indicates that multi-scale statistics pooling outperforms native statistics pooling in the networks with ISKConv. Nonetheless, ResNet34+MSSP does not perform better than ResNet34 significantly, demonstrating that MSSP only work well with the combination of the ISKConv. Then we replace ISKConvs of the proposed architecture with 3×3 convolutions to obtain a ResNet34+MSSP model. Compared with the proposed network, the ResNet34+MSSP achieves higher EERs of 17.7%, 18.2% and 14.2% and higher MinDCFs of 25.0%, 13.2% and 11.6%, which demonstrates the effectiveness of ISKConv. In addition, ResNet34+ISKConv significantly outperforms ResNet34 when the networks are trained with variable-length training samples, indicating that the proposed improved selective kernel convolution is fit for the variable-length training technique, which is also verified in TDNN-based TI-SV systems [25].

We also compare the proposed system with recently reported CNN-based TI-SV systems in terms of performance on VoxCeleb1-O test set, as shown in Table 3. Note that ResNet34+SP [4] has two versions of embeddings. We select the 256-dimensional embeddings as same as our architecture for fair comparison. The proposed network achieves the best EER of 1.292% as well as MinDCF of 0.1618 among all systems without any data augmentation techniques.

Table 3: *Comparison with recently reported CNN-based systems in VoxCeleb1-O test set.* $*$ *means using data augmentation and NR means "not reported".*

Systems	Train set	EER(%)	MinDCF
VGG-M [12]	Vox1	7.8	0.71
ResNet34+LDE [26]	Vox1	4.56	0.441
Thin-ResNet34+GhostVLAD [8]	Vox2	3.22	NR
ResNet50+EAMS [9]	Vox2	2.94	0.278
ResNet34+SPE [27]	Vox2	2.61	0.245
PRN-50v2+ft-CBAM [7]	Vox2	2.03	NR
ResNet34+LDE+FPMTC [10]	Vox2	1.98	0.205
BLSTM-ResNet [28]	Vox2	1.87	NR
ResNet34+SP [4]	Vox2*	1.42	0.166
ResNet34+ISKConv+MSSP	Vox2	**1.292**	**0.1618**

5. Conclusions

In this paper, we investigate the effect of using variable-length training samples on deep CNN-based TI-SV systems and explore two approaches for capturing variable-term contexts to improve the performance of deep CNN architectures: improved selective kernel convolution and multi-scale statistics pooling. Experiments are conducted on VoxCeleb datasets. According to the experimental results, we obtain two conclusions as follows: (1) The effect of using variable-length training samples is diverse in different CNN architectures. (2) The constructed architecture with two proposed methods achieves significant improvement over r-vectors baseline system.

6. Acknowledgements

This work is supported by National Natural Science Foundation of China (Grant No. 62002177), Science and Technology Planning Project of Tianjin, China (Grant No. 20YDT-PJC01810), Tianjin Natural Science Foundation (Grant No. 19JCQNJC00300) and Research and Innovation Project for Postgraduates in Tianjin (Artificial Intelligence) (Grant No. 2020YJSZXB01).

7. References

[1] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[2] S. Ioffe, "Probabilistic linear discriminant analysis," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 531–542.

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

[4] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "But system description to voxceleb speaker recognition challenge 2019," 2019.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] J. S. Chung, A. Nagrani, E. Coto, W. Xie, M. McLaren, D. A. Reynolds, and A. Zisserman, "Voxsrc 2019: The first voxceleb speaker recognition challenge," *ISCA Challenges*, 2019.

[7] S. Yadav and A. Rai, "Frequency and temporal convolutional attention for text-independent speaker recognition," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6794–6798.

[8] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5791–5795.

[9] Y. Yu, L. Fan, and W. Li, "Ensemble additive margin softmax for speaker verification," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6046–6050.

[10] Y. Jung, S. M. Kye, Y. Choi, M. Jung, and H. Kim, "Improving multi-scale aggregation using feature pyramid module for robust speaker verification of variable-duration utterances," in *Proc. Interspeech 2020*, 2020, pp. 1501–1505.

[11] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[12] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.

[13] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[14] Z. Gao, Y. Song, I. McLoughlin, W. Guo, and L. Dai, "An improved deep embedding learning method for short duration speaker verification," in *Proc. Interspeech 2018*, 2018, pp. 3578–3582.

[15] L. You, W. Guo, L.-R. Dai, and J. Du, "Deep neural network embeddings with gating mechanisms for text-independent speaker verification," in *Proc. Interspeech 2019*, 2019, pp. 1168–1172.

[16] Y. Wu, C. Guo, H. Gao, J. Xu, and G. Bai, "Dilated residual networks with multi-level attention for speaker verification," *Neurocomputing*, vol. 412, pp. 177–186, 2020.

[17] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 1451–1460.

[18] Z. Gao, Y. Song, I. McLoughlin, P. Li, Y. Jiang, and L.-R. Dai, "Improving aggregation and loss function for better embedding learning in end-to-end speaker verification system," in *Proc. Interspeech 2019*, 2019, pp. 361–365.

[19] S. Seo, D. J. Rim, M. Lim, D. Lee, H. Park, J. Oh, C. Kim, and J.-H. Kim, "Shortcut connections based deep speaker embeddings for end-to-end speaker verification system," in *Proc. Interspeech 2019*, 2019, pp. 2928–2932.

[20] A. Hajavi and A. Etemad, "A deep neural network for short-segment speaker recognition," in *Proc. Interspeech 2019*, 2019, pp. 2878–2882.

[21] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," 2011.

[22] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," in *Proc. Interspeech 2019*, 2019, pp. 2873–2877.

[23] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[24] P. Matějka, O. Novotný, O. Plchot, L. Burget, M. D. Sánchez, and J. Černocký, "Analysis of score normalization in multilingual speaker recognition," in *Proc. Interspeech 2017*, 2017, pp. 1567–1571.

[25] Y.-Q. Yu and W.-J. Li, "Densely connected time delay neural network for speaker verification," in *Proc. Interspeech 2020*, 2020, pp. 921–925.

[26] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 74–81.

[27] Y. Jung, Y. Kim, H. Lim, Y. Choi, and H. Kim, "Spatial pyramid encoding with convex length normalization for text-independent speaker verification," in *Proc. Interspeech 2019*, 2019, pp. 4030–4034.

[28] Y. Zhao, T. Zhou, Z. Chen, and J. Wu, "Improving deep cnn networks with long temporal context for text-independent speaker verification," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6834–6838.