ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414765

# TIME-DOMAIN SPEAKER VERIFICATION USING TEMPORAL CONVOLUTIONAL NETWORKS

*Sangwook Han, Jaeuk Byun, and Jong Won Shin*

School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology, Gwangju, Korea

## ABSTRACT

Recently, speaker verification systems using deep neural networks have been widely studied. Many of them utilize hand-crafted features such as mel-filterbank energies, mel-frequency cepstral coefficients, and magnitude spectrograms, which are not designed specifically for the speaker verification task and may not be optimal. Recent releases of the large datasets such as VoxCeleb enable us to extract the task-specific features in a data-driven way. In this paper, we propose a speaker verification system that takes the time-domain raw waveforms as inputs, which adopts a learnable encoder and temporal convolutional networks (TCNs) that have shown impressive performance in speech separation. Moreover, we have applied the squeeze and excitation networks after each TCN block to apply channel-wise attention. Our experiments on the VoxCeleb1 dataset demonstrate that the speaker verification system utilizing the proposed feature extraction model outperforms previously proposed time-domain speaker verification systems.

***Index Terms*—** Speaker verification, text-independent, data-driven, time-domain, attention.

## 1. INTRODUCTION

Speaker verification (SV) refers to the task to authenticate the identity claim of a speaker based on that speaker's known utterances. With the advances in deep learning, the systems that extract speaker embeddings using deep neural networks (DNNs) [1, 2] have achieved better performances compared with conventional i-vector-based systems [3]. A variety of neural networks including ResNet34 [4] and time delay neural network (TDNN) [5] have proven to produce robust speaker embedding. Furthermore, various pooling methods have been proposed to aggregate frame-level features into an utterance-level representation. Among them, the attention mechanisms [6, 7, 8, 9] that weigh the important frames have been shown to be effective. In the meanwhile, a variety of loss functions for discriminative speaker embedding have been studied in [10]. Many recent studies, however, have used the magnitude spectrogram or hand-crafted features such as mel-filterbank energies, and mel-frequency cepstral coefficients (MFCCs), which are not specialized for the speaker verification task. In [11], speaker embeddings are directly extracted from raw waveform using convolutional neural network (CNN)-gated recurrent unit (GRU). [12] proposed a model that captures the important speaker characteristics from waveforms better by replacing the first convolutional layer with a parameterized layer of the sinc functions.

Recently, the time-domain speech separation approaches have shown impressive performances, in which the short-time Fourier transform (STFT) is replaced with a data-driven transformation [13]. [13] applied the temporal convolutional networks (TCNs) between the learnable encoder and decoder for efficient modeling of temporal context. Inspired by the [13], we propose a time-domain speaker verification system which adopts a learnable encoder and TCNs. Moreover, we propose the 1-D ConvSE block that combines a TCN block with the squeeze and excitation network (SENet) [14], which enables to account for the temporal context while focusing on the important channels.

This paper is organized as follows. Section 2 describes the proposed system and 1-D ConvSE blocks. Our experimental setup and results are given in Section 3 and 4. The last section concludes the paper.

## 2. PROPOSED SYSTEM

The overall structure of the proposed speaker verification system is described in Fig. 1. The system consists of the feature extraction (FE), speaker model, and classification backends. For the backends, the classifier that produces the logit tensor for speaker identification is connected during the training phase, while the similarity measure-based speaker verification is performed during the evaluation after removing the classifier. The details of each stage will be introduced in the following subsections. For brevity, we denote the convolution having the kernel size and stride of 1 to the $1\times1$-*conv*.

This research was funded by National Research Foundation of Korea grant number NRF-2019R1A2C2089324 and Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2019 (R2019080018).

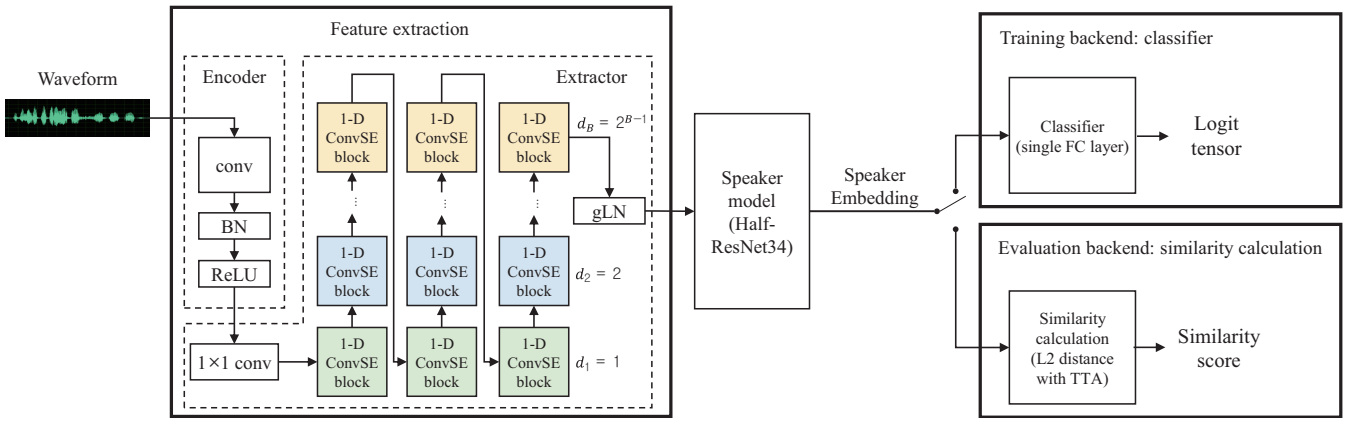


**Fig. 1**. Overall structure of the proposed speaker verification system.

### 2.1. Encoder

The features based on the knowledge for the nature of the speech signal such as the magnitude of STFT coefficients, mel-filterbank energies, and MFCCs have been widely used for speaker verification, but they are not designed specifically for the given task and may not be optimal. Inspired by [13], we adopt a learnable encoder which uses the convolutional kernels that can be learned using the loss function specialized for speaker recognition. Specifically, the 1-D convolutional layer using the $H$ kernels with the length $L$ and the stride of $L/2$ encodes the raw waveform input $x \in \mathbb{R}^{1\times T}$ to the representation $\mathbf{X} \in \mathbb{R}^{H\times K}$, where $T$ denotes the input length and $K = 2(T-L)/L + 1$. And then, batch normalization (BN) and rectified linear units (ReLUs) follow for stable training.

### 2.2. Extractor

At the beginning of the extractor module, the output of encoder is passed through a 1×1-*conv* with $P$ filters to adjust the channel size for the subsequent blocks. The remaining part of extractor consists of TCN blocks combined with the SENet [14]. The structure of the extractor is basically similar to [15], which consists of the $R$ repetitions of the $B$ stacked TCN blocks whose dilation factor $d$ increases exponentially as $d_i = 2^{i-1}, i = 1, \ldots, B$.

The TCN block used in [15] consists of 1×1-*conv* with $M$ filters and the depthwise convolution (*D-conv*) where each of them is followed by a parametric ReLU (PReLU) and BN. Then, it is compressed to $P$ dimension using 1×1-*conv* and summed with the block input for the residual modeling. Here, we have added the channel-wise attention after the last 1×1-*conv* in the original TCN block using the SENet as depicted in Fig. 2, which is denoted as the 1-D ConvSE block. The flowchart of the SENet in the 1-D ConvSE block is depicted in Fig. 3, in which the input to the SENet is denoted as $\mathbf{W} \in \mathbb{R}^{P\times K}$ and the output is $\widetilde{\mathbf{W}} \in \mathbb{R}^{P\times K}$. The SENet

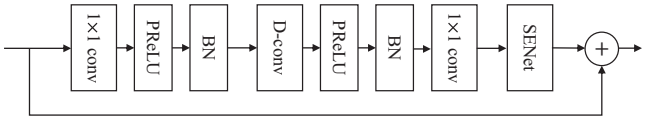


**Fig. 2**. The structure of 1-D ConvSE block.

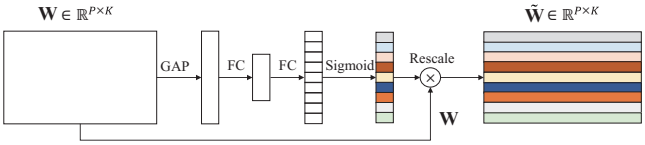


**Fig. 3**. The flowchart of SENet.

first aggregates each channel by the global average pooling (GAP) followed by the bottleneck with two fully connected (FC) layers. The first FC layer reduces the dimension by the factor of $r$ with the ReLU activation, and the second FC layer expands back to the original dimension with the sigmoid activation. Then, the resulting vector works as a weight for the input channels which is applied to all columns in $\mathbf{W}$. By using the stacked 1-D ConvSE blocks, the receptive field gets large so that long-range dependencies of the input waveform can be captured, while channel-wise attention is applied for each TCN with a different dilation factor. Then, the output of the last 1-D ConvSE block is normalized over the channel and time dimensions using global layer normalization (gLN) [13]. The feature extraction by our stacked 1-D ConvSE blocks will thereby efficiently work as a pre-processing for the speaker model that we will elaborate in the next subsection.

### 2.3. Speaker model

The speaker model uses the extracted features to produce speaker embeddings that are used for speaker identification

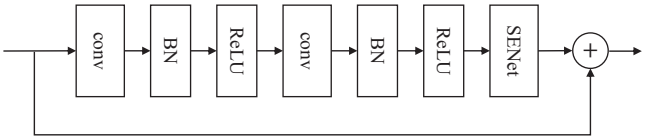


**Fig. 4**. The structure of ResSE block in the speaker model.

in the training phase and the speaker verification in the test phase. We use a modified version of the ResNet34 [4] in which the number of channels in each layer is a half of that in the original ResNet34 and 2-D convolutions are replaced by 1-D convolutions as the speaker model. Additionally, we have added the SENet inside of the blocks in the ResNet as in the case of the 1-D ConvSE block. Our speaker model is denoted as the Half-ResNet34, which is basically the stacked ResSE blocks shown in the Fig.4. Each ResSE block consists of two convolutional layers, each of which is followed by the BN and ReLU activation, the additional SENet layer after the second ReLU, and the residual connection. In [6] and [10], the Thin-ResNet in which the number of channels in each layer is a quarter of that in the ResNet34 was used as the speaker model, but we choose the Half-ResNet34 to prevent too much dimension reduction at the first layer of the speaker model. After all the ResSE blocks, we aggregate the frame-level features over the temporal dimension to produce the utterance-level speaker information using the self-attentive pooling (SAP) [6]. After that, the last FC layer yields the speaker embedding vectors.

### 2.4. Classification backends

To obtain the discriminative speaker embeddings, we first train our end-to-end model for the closed-set speaker identification task using a classifier with a single FC layer. After the training, Euclidean distance between the speaker embedding vectors from the enrollment and verification utterances is calculated as a similarity measure, replacing the classifier used in the training. Here, we apply the test-time augmentation (TTA) method [16], which divides each utterance into equally-spaced segments and calculates the average of the similarities for all possible combinations of segment pairs as a final score.

### 2.5. Loss function

We used the softmax and additive margin softmax (AM-Softmax) [17] loss functions for the end-to-end training of the proposed speaker verification system. [17] has reported that AM-Softmax better minimizes the intra-class variance while maximizing the inter-class variance than softmax by introducing an additive margin in the angular space. The AM-softmax loss function is defined as

| Symbol | Description | Value |
|---|---|---|
| $T$ | Length of segments | 65,536 |
| $H$ | Number of channels in encoder | 512 |
| $P$ | Number of channels for input/ouptut to ConvSE | 128 |
| $M$ | Number of channels in ConvSE | 256 |
| $B$ | Number of convolutional blocks | 8 |
| $R$ | Number of repeats | 3 |
| $r$ | Reduction ratio | 16 |

**Table 1**. Hyperparameters of the proposed system.

| Filter length ($L$) | ConvSE | EER(%) |
|---|---|---|
| 64 | × | 5.07 |
| 64 | ○ | 4.51 |
| 40 | × | 4.64 |
| 40 | ○ | 4.15 |

**Table 2**. EERs for the proposed system according to the filter length in the encoder and the use of 1-D ConvSE blocks.

$$L_{AMS} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(cos(\theta_{y_i,i})-m)}}{\sum_{j\neq y_i} e^{s(cos(\theta_{j,i})} + e^{s(cos(\theta_{y_i,i})-m)}} \quad (1)$$

where $N$ is the number of utterances in a mini-batch, $y_i$ is the speaker label corresponding to the $i$-th utterance, $cos(\theta_{j,i})$ is the dot product of the normalized weight vector for the $j$-th candidate speaker in the FC layer of the classifier and the normalized speaker embedding which is the input to the FC layer, and $m$ and $s$ refer to an additive margin and a scale factor, respectively.

## 3. EXPERIMENTS

### 3.1. Datasets

To evaluate the performance of the proposed model, we used the VoxCeleb1 dataset [18], of which the training set contains 148,642 utterances from 1,211 speakers for text-independent speaker verification and the test set includes 4,874 utterances spoken by 40 speakers. The speaker verification performance was evaluated with 37,720 verification-enrollment pairs comparing 4,715 utterances with 4 positive (same speaker) and 4 negative (different speaker) utterances. For both training and test data, the pre-emphasis and the mean and variance normalization are applied on the 16 kHz audio files.

### 3.2. Implementation detail

We randomly cropped a 4.096 s-long (= 65,536 samples) segment from each utterance as an input to the network for every

| | Input Feature | Front-end | Loss | Dims | Aggregation | EER(%) |
|---|---|---|---|---|---|---|
| Nagrani *et al*. [18] | Spectrogram | VGG-M | Softmax | 1024 | TAP | 10.20 |
| Hajibabaei *et al*. [19] | | ResNet20 | Softmax | 128 | TAP | 6.73 |
| Hajibabaei *et al*. [19] | | ResNet20 | A-Softmax | 128 | TAP | 4.40 |
| Hajibabaei *et al*. [19] | | ResNet20 | AM-Softmax | 128 | TAP | 4.30 |
| Cai *et al*. [6] | Mel-filterbank energy | Thin-ResNet34 | A-Softmax | 128 | SAP | 4.40 |
| Wang *et al*. [9] | | ResNet34 | AM-Softmax | 512 | MRMHA | 3.96 |
| Gao *et al*. [20] | | ResNetXt | DALoss | 512 | Statistical | 3.87 |
| Okabe *et al*. [7] | MFCC | TDNN(x-vector) | Softmax | 1500 | ASP | 3.85 |
| Jung *et al*. [11] | Raw waveform | RawNet | Softmax | 128 | GRU | 6.80 |
| Jung *et al*. [11] | | RawNet | Softmax+Center+BS | 128 | GRU | 4.80 |
| **Ours** | | FE + Half-ResNet34 | Softmax | 256 | SAP | 5.76 |
| **Ours** | | FE + Half-ResNet34 | AM-Softmax | 256 | SAP | **4.15** |

**Table 3**. Performances for recently proposed systems on the test set of VoxCeleb1, trained with the development set of VoxCeleb1. TAP: temporal average pooling, SAP: self-attentive pooling, A-Softmax: angular softmax, MRMHA: multi-resolution multi-head attention, DALoss: discriminant analysis loss [20], ASP: attentive statistics pooling, and BS: speaker basis loss [21].

epoch. The Adam optimizer was used with the learning rate which was initialized as 0.001 and decayed 15% for every 10 epochs. The value of $s$ was fixed at 30, while an annealing strategy on $m$ was applied to stabilize the training. $m$ was initially set to 0 and linearly increased by 0.01 per 10 epochs. The hyper-parameters used in our model are summarized in Table 1. The kernel size of *D-conv* in 1-D ConvSE block was set to 3. Other pre-processings such as voice activity detection (VAD) and data augmentation were not applied. We trained the model for 250 epochs and then used the resultant model for evaluation. The performances were measured in terms of the equal error rate (EER, %), which is the false acceptance rate (FAR) when it is the same as the false rejection rate (FRR). Our Pytorch implementation is available at [1].

## 4. RESULTS

Table 2 shows the EERs for the proposed model with respect to the filter length in the encoder and the use of SENet in the TCN blocks in the feature extractor. As we can see, the use of the channel-wise attention via SENet was beneficial for speaker verification. Also, we see that using shorter window length showed better EER performance at the cost of increased computational complexity.

Table 3 summarizes the performances of the proposed and the recently proposed speaker verification systems trained with the training set of the VoxCeleb1 database and tested on the VoxCeleb1 test set. To focus on the speaker embedding construction, we excluded the systems with deep learning-based speaker verification backend such as what is reported in [11]. Within the systems using the same input features, [19], [20], [7], and the proposed system performed the best for spectrogram, mel-filterbank energy, MFCC, and raw waveform, respectively. Among them, [7] using MFCCs as input features demonstrated the best performance, although we expected the learnable transformation would result in better performance than hand-crafted features when end-to-end training was applied. Still, the performance of the proposed speaker verification using raw waveforms may be improved if the data augmentation, aggregation methods, and loss functions used in compared models are adopted, as they can be applied to the proposed system independently to our main contribution on the feature extraction part. The comparison among the time-domain speaker verification systems showed that our proposed system resulted in the best EER of 4.15 % with 3.83M model parameters, while the method in [11] reported to have 4.8 % of EER with 5.78M parameters when the same speaker verification backend was applied.

## 5. CONCLUSIONS

In this paper, we propose a time-domain speaker verification system, which adopts the learnable encoder and temporal convolutional networks (TCNs) that achieved a great success in the speech separation. Moreover, we have applied channel-wise attention at the end of each TCN block in the feature extractor and the ResBlock in the speaker model in the form of the SENet to further enhance the performance. Although the proposed model did not show better performance than the existing systems using the hand-crafted features, it demonstrated the best performance among the time-domain approaches without sophisticated backend and obtained the competitive EER of 4.15 % implying the potential for extracting features directly from the waveform using deep learning.

[1] https://github.com/blackcoWook/time-domain-SV

## 6. REFERENCES


[1] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[2] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.

[3] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] 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.

[6] Weicheng Cai, Jinkun Chen, and Ming 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.

[7] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda, "Attentive statistics pooling for deep speaker embedding," *Proc. Interspeech 2018*, pp. 2252–2256, 2018.

[8] Pooyan Safari and Javier Hernando, "Self multi-head attention for speaker recognition," *Proc. Interspeech 2019*, pp. 4305–4309, 2019.

[9] Zhiming Wang, Kaisheng Yao, Xiaolong Li, and Shuo Fang, "Multi-resolution multi-head attention in deep speaker embedding," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6464–6468.

[10] Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han, "In defence of metric learning for speaker recognition," *arXiv preprint arXiv:2003.11982*, 2020.

[11] Jee-weon Jung, Hee-Soo Heo, Ju-ho Kim, Hye-jin Shim, and Ha-Jin Yu, "Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification," *Proc. Interspeech 2019*, pp. 1268–1272, 2019.

[12] Mirco Ravanelli and Yoshua Bengio, "Speaker recognition from raw waveform with sincnet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 1021–1028.

[13] Yi Luo and Nima Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[14] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[15] Yi Luo and Nima Mesgarani, "Tasnet: Surpassing ideal time-frequency masking for speech separation," *arXiv preprint arXiv:1809.07454*, 2018.

[16] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," *Proc. Interspeech 2018*, pp. 1086–1090, 2018.

[17] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[18] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: A large-scale speaker identification dataset," *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

[19] Mahdi Hajibabaei and Dengxin Dai, "Unified hypersphere embedding for speaker recognition," *arXiv preprint arXiv:1807.08312*, 2018.

[20] Zhifu Gao, Yan Song, Ian McLoughlin, Pengcheng Li, Yiheng Jiang, and Li-Rong Dai, "Improving aggregation and loss function for better embedding learning in end-to-end speaker verification system.," in *INTERSPEECH*, 2019, pp. 361–365.

[21] Hee-Soo Heo, Jee-weon Jung, IL-Ho Yang, Sung-Hyun Yoon, Hye-jin Shim, and Ha-Jin Yu, "End-to-end losses based on speaker basis vectors and all-speaker hard negative mining for speaker verification," *Proc. Interspeech 2019*.