IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 22, 2019, accepted May 6, 2019, date of publication May 24, 2019, date of current version July 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2917470

Deep CNNs With Self-Attention for Speaker Identification

NGUYEN NANG AN[1], NGUYEN QUANG THANH[1], AND YANBING LIU[2]
[1]Department of Computer Science and Technology, Chongqing University of Posts and Telecommunication, Chongqing 400065, China
[2]Chongqing Engineering Laboratory of Internet and Information Security, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Corresponding author: Nguyen Nang An (3216801919@qq.com)

ABSTRACT Most current works on speaker identification are based on i-vector methods; however, there is a marked shift from the traditional i-vector to deep learning methods, especially in the form of convolutional neural networks (CNNs). Rather than designing features and a subsequent individual classification model, we address the problem by learning features and recognition systems using deep neural networks. Based on the deep convolutional neural network (CNN), this paper presents a novel text-independent speaker identification method for speaker separation. Specifically, this paper is based on the two representative CNNs, called the visual geometry group (VGG) nets and residual neural networks (ResNets). Unlike prior deep neural network-based speaker identification methods that usually rely on a temporal maximum or average pooling across all time steps to map variable-length utterances to a fixed-dimension vector, this paper equips these two CNNs with a structured self-attention mechanism to learn a weighted average across all time steps. Using the structured self-attention layer with multiple attention hops, the proposed deep CNN network is not only capable of handling variable-length segments but also able to learn speaker characteristics from different aspects of the input sequence. The experimental results on the speaker identification benchmark database, VoxCeleb demonstrate the superiority of the proposed method over the traditional i-vector-based methods and the other strong CNN baselines. In addition, the results suggest that it is possible to cluster unknown speakers using the activation of an upper layer of a pre-trained identification CNN as a speaker embedding vector.

INDEX TERMS Speaker identification, deep neural networks, self-attention, embedding learning.

I. INTRODUCTION

Speaker identification has gained increasing attention from the academic and industry communities in recent years [1]–[3], and it is been widely used in applications, including *surveillance* [4], *discriminative speaker embedding learning* [5]–[7], and *speaker diarization* [8]. The principal goal of speaker identification is to automatically infer the identity of a speaker from an input utterance given a closed set of known voice models [1]–[3], [9]. Generally, a traditional speaker identification system starts with acoustic feature extraction, such as mel-frequency cepstrum coefficients (MFCCs), and then utilizes a large scale of unlabeled speech data to train a model to capture speaker characteristics in an unsupervised way, finally training a classifier for the speaker classification.

A large number of various signal processing and pattern recognition methods have been successfully applied to the speaker identification task, which include wavelet [10], hidden markov models (HMMs) [11], [12], vector quantization (VQ) [12], [13], sparse coding [14], Gaussian mixture models (GMMs) [11], Gaussian mixture model-universal background models (GMM-UBM), i-vector [15], support vector machines (SVMs) [16], and, most recently deep neural networks [17]–[20]. Specifically, the classic GMM-based method [11] was inspired by the interpretation that the Gaussian components represent some general speaker-dependent spectral shapes and the capability of Gaussian mixtures to model arbitrary densities. Moreover, Campbell *et al.* [16] approached the problem by using SVMs to map inputs into a high-dimensional space and then separating classes with a hyper-plane. Afterwards, Campbell *et al.* [21] further combined SVMs with the GMM supervector concept. In this approach, GMMs are used for latent factor analysis to

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato.

compensate for the speaker and the channel [22], leading to a GMM supervector of the stacked means of the mixture components. Then the SVM model takes the resulting GMM supervector as input to build a classifier.

The i-vector systems, which use GMM factor analysis, are aimed at learning to compensate for both the speaker and channel variability in a low-dimensional (e.g., a few hundred) space, normally called the total variability subspace [15]. In addition, universal back-ground models (UBMs) are employed to generate frame-level soft alignments required in the i-vector estimation process. The i-vectors are typically post-processed through a linear discriminant analysis (LDA) [23] stage to generate dimensionality-reduced and channel-compensated features, which can then be efficiently modeled and scored with various classification backends such as a probabilistic LDA (PLDA) or an SVM [5], [24], [25], resulting in a hybrid system. In conclusion, because of the great success of these aforementioned methods, the i-vector speaker recognition hybrid systems still dominate most of the current research on speaker identification [15], achieving the best performance in recent NIST (National Institute of Standards and Technology) evaluations of both speaker and language recognition [26]. However, these hybrid systems are problematic since they are designed to train the different modules separately with different criteria, which may not be optimal for the final speaker identification task. In contrast, we propose an end-to-end speaker identification framework, in which we combine CNNs with the structured self-attention mechanism without intermediate models.

Recently, with the increase in deep learning in the speech recognition community, a number of various deep neural networks (DNNs) have been successfully applied to speaker recognition. Lei et al. in [27] proposed a method using a phonetically aware deep neural network method for speaker recognition, where the DNNs first replace the standard GMM to produce frame alignments and then use it to enhance phonetic modeling in the i-vector UBM. The system is highly dependent on the need for transcribed in-domain training data and greatly increases the computational complexity. More recently, the design of end-to-end DNN-based speaker recognition systems is currently a very active research area, which can be directly optimized to discriminate between different speakers [5]–[7], [19], [28]. This has the potential to produce efficient, compact and scalable systems, which only require speaker labels for training and are capable of leveraging large amounts of data to capture the characteristics of the speaker. The early systems often apply DNNs to separate speakers, leading to frame-level feature representations, which are then used as input to Gaussian speaker models [29], [30]. Heigold et al. introduced an end-to-end system, trained on the phrase ''OK Google'' that jointly learns an embedding along with a similarity metric to compare pairs of embeddings [28]. Snyder et al., introduced a temporal pooling layer into a DNN to map variable-length utterances to fixed-dimension embeddings for a text-independent application [31]. Further, Snyder et al. in [19] systematically investigated the impact of data augmentation techniques.

Nevertheless, existing methods simply ignore leveraging the temporal information to compute fixed speaker embeddings; therefore, a model with temporal information, has not yet been implemented in speech processing. In this paper, to leverage the temporal information for speaker embeddings, we propose the insertion of the structured self-attention layer into a CNN for text-independent speaker identification. Deep CNNs equipped with the structured self-attention mechanism are very well suited to the problem identification. First, deep CNNs are capable of capturing energy modulation patterns across time and frequency when applied to spectrogram-like inputs, which have been shown to be an important trait for distinguishing among different speaker characteristics. Second, by using the structured self-attention layer with multiple attention hops, the network is able to not only handle variable-length segments, but also learn speaker characteristics from different aspects of the input sequence, which explicitly exploits the temporal or context knowledge to form speaker embeddings. Analogous to the works in [5], [6], the proposed system is systematically evaluated on the publicly available VoxCeleb database, which is a large-scale text-independent speaker identification corpus.

The organization of this paper is as follows. Section III first presents the proposed methods for speaker identification, which include two representative deep CNNs, the self-attention algorithm, and the objective function. Then, in Section IV, we show the experimental results on the VoxCeleb database. Finally, Section V concludes this paper and suggests directions for the future work.

II. RELATED WORK

There is currently a large body of research regarding the use of deep CNNs for audio data. Common use cases are efficiently recognizing text or emotional states from speech, as well as differentiating acoustic events in a large corpus [32]–[35]. These systems are based on low-level acoustic features such as MFCCs and use a variety of deep CNNs techniques, e.g., VGG nets [36], ResNets [37], and the Inception network [38]. Although these approaches work well for speech recognition and acoustic event detection, there are a handful of related works on text-independent speaker identification, and hence, they require a more sophisticated acoustic model.

A recent paper [27] employs DNNs for speaker recognition with phoneme detection, in which, based on the i-vector framework, DNNs are only used to replace the standard GMM to yield frame alignments to enhance phonetic modeling in the i-vector UBM. Moreover, although the authors have proven the efficiency on the 2012 NIST speaker recognition evaluation, it is primarily intended for text-dependent speaker authentication.

The most closely related works are [5], [6] in which the authors also make use of deep CNNs to directly optimize

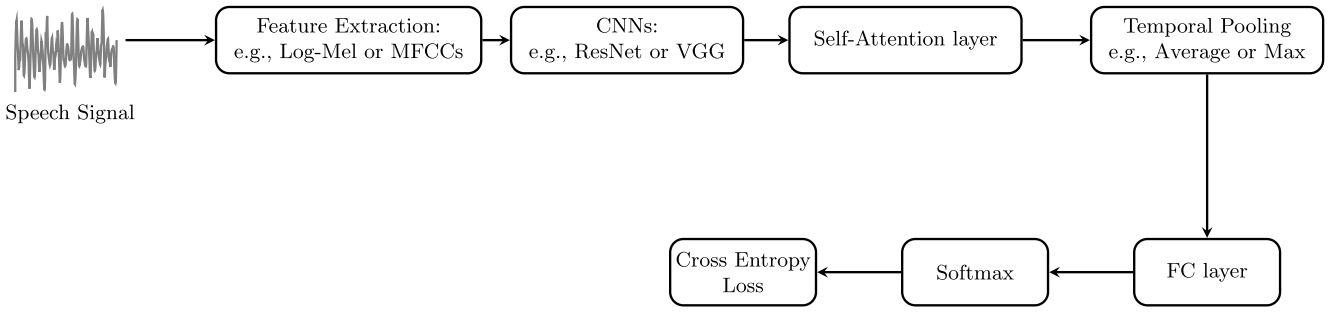

FIGURE 1. Overview of the proposed system for closed-set speaker identification. Note that while MFCCs correspond to mel-frequency cepstrum coefficients, FC corresponds to fully connected layers.

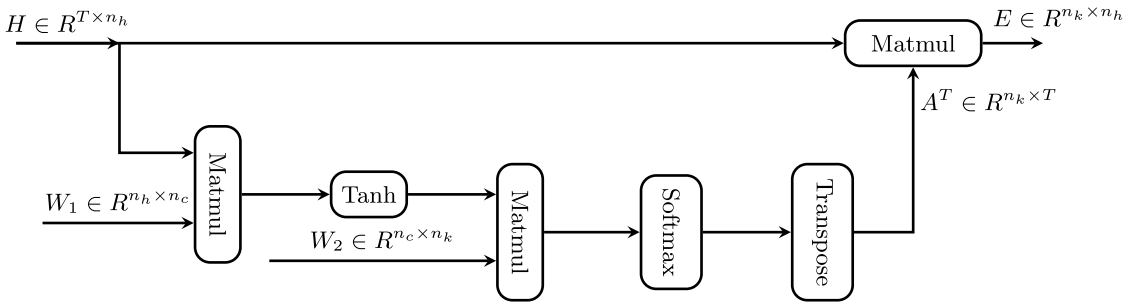

FIGURE 2. Computation graph of the self-attention layer used in the proposed network.

them for the text-independent speaker identification task in an end-to-end way. Most importantly, the authors also adapt the VGG CNN network and the ResNets for the 1251-way speaker identification problem on the VoxCeleb database. However, the temporal average pooling mechanism or the temporal maximum pooling mechanism is adopted to form a fixed speaker embedding. Though the evaluation process on the VoxCeleb database has presented an obvious performance gain over the i-vector speaker identification method, such methods ignore the temporal information implied by the speech signals, which has been proven helpful for speaker identification [1]–[3]. In contrast, we use the self-structured attention mechanism, originally introduced in [39] for sentence embedding, which is capable of exploiting the temporal knowledge in processing sequential data(eg, speech signals) and provides provable guarantees in conjunction with deep CNNs. In our evaluation, we strictly follow the evaluation process defined in [5], and the evaluation results demonstrate that our proposed method significantly outperforms that proposed in [5].

III. PROPOSED METHODS

A. SPEAKER IDENTIFICATION SYSTEM DESCRIPTION

The closed-set speaker identification system this paper focuses on can be viewed as a multi-class classification problem. That is, given a test utterance, such an identification system assigns a speaker label in the set of registered speakers. Inspired by the great success of deep neural networks in speech recognition, speech emotion recognition, sound event detection, and image classification, our proposed system is rooted in two representative deep CNNs, the VGG CNN [36] and ResNets [37], known for their great classification performance in large-scale image classification tasks and speech recognition tasks [34], [35]. On top of the VGG-like CNN and ResNets, there is one structured self-attention layer [39], followed by a temporal average pooling layer. Similar to the common structure of DNNs for a classification task, the topmost layer is a softmax layer. The structure of the proposed network is shown in Figure 1.

B. SELF-ATTENTION MECHANISM

The self-attention mechanism has recently become popular because it has been successfully applied in several tasks, including speech recognition [40]–[42], speech emotion recognition [43], phoneme recognition [44], and neural machine translation [45], [46].

In this paper, we adopt the structured self-attention layer originally introduced in [39] for sentence embedding. Figure 2 illustrates the computation graph of the structured self-attention layer. Given a speech sequence of T frames $H = (h_1, h_2, \ldots, h_T)$ that have the size T-by-n_h and are the hidden outputs from the previous layer, the self-attention layer performs a series of linear combinations of the T hidden vectors in H with the ultimate aim of encoding such a variable length sequence into a fixed-size embedding matrix. Specifically, the self-attention layer first takes the entire hidden outputs H as input and computes an annotation matrix of weights A as follows:

$$A = \mathtt{softmax}(\mathtt{tanh}(HW_1)W_2), \tag{1}$$

where $W_1 \in R^{n_h \times n_c}$ and $W_2 \in R^{n_c \times n_k}$ are two trainable matrices, and the annotation matrix A will be sized by T-by-n_k, and $\texttt{tanh}(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$ is the hyperbolic tangent function. The hyper-parameter n_k corresponds to the number of attention hops. Here, the $\texttt{softmax}()$ is applied along the first dimension of its input, which ensures the computed weights sum up to one.

Finally, a linear map is again used to mix the different h_t from different n_k attention hops, leading to speaker embedding E

$$E = A^T H, \tag{2}$$

where E is a matrix of a shape of $n_k \times n_h$. Table 1 lists all the variable notations. Note that the structured self-attention layer can be eventually implemented using a two-layer feed-forward neural network, which allows fast computation.

TABLE 1. Notations introduced in the self-attention algorithm.

Abbreviations	Explanations
A	Annotation matrix
H	A sequence with the size T-by-n_h
T	The sequence length
W_1	A trainable matrix with size n_h-by-n_c
W_2	A trainable matrix with size n_c-by-n_k
$\texttt{tanh}(x)$	The hyperbolic tangent function
n_c	The hidden size of the middle layer
n_k	The number of attention hops
n_h	The dimension of the each frame in the sequence
E	The speaker embedding
$\texttt{softmax}(x)$	The softmax function
I	Identity matrix
β	Tuning parameter

When compared with the standard self-attention mechanism only using one hop that usually focuses on a very specific area of the input speech sequence [6], the structured self-attention mechanism utilizes n_k attention hops to learn to capture the essential speaker characteristics from multiple areas of the input speech sequence. However, as the number of the attention hops n_k increases, the attention matrix A tends to suffer from redundancy problems [39]. To address the redundancy problem, as suggested in [39], a penalization term is added to the loss of the network (cf. Section III-E), which is defined by

$$l_p = \beta \|\|A^T A - I\|\|_F^2, \tag{3}$$

where $\|\|\cdot\|\|$ is the Frobenius norm of a matrix, I is the identity matrix with a shape of $n_k \times n_k$, and $\beta > 0$ is a tuning parameter controlling the importance of the penalization term.

C. VGG-LIKE CNNS

Based on the structured self-attention layer, we first propose to combine it and the VGG convolution net to form a speaker identification system. The deep VGG convolution net was originally proposed for image classification in the ImageNet 2014 competition [36]. Since then, the VGG-inspired networks have been successfully adapted to image classification [36], ASR [32], [33], large-scale audio classification [34], and speech emotion recognition [35]. The basic concept of the VGG is to construct a multiple-layer convolution network by using small 3×3 convolutional kernels with rectified linear unit ($ReLU$) and non-linear functions without pooling (eg, max or average) between these layers. Here, we make use of this concept to construct a VGG-like CNN, which consists of 7 hidden convolutional layers, one structured self-attention layer, one temporal average pooling layer, and two fully connected layers. Note that the structured self-attention layer not only generates a fixed length input for the following fully connected layers but also allows for the model to jointly attend to discriminative speaker information from different positions. Further, to reduce the computational burden of the following fully connected layers, the temporal average pooling layer is applied to the resulting speaker embedding. The ReLU activation is used for each hidden layer. Additionally, batch normalization [47] is applied for each convolutional layer. The full details on the proposed VGG-like CNN are given in Table 2, and Figure 3 illustrates the structure of the modified VGG network.

TABLE 2. Modified VGG architecture with the self-attention layer and an average pool layer at the end for speaker identification. The ReLU and batch normalization layers are not shown. Each row specifies the number of convolutional filters and their sizes as filter_size × filter_size, # filters. Here the self-attention layer corresponds to the layer introduced in Section III-B.

Layer	VGG-like CNN	Output ($T \times F \times C$)
Input:	–	$300 \times 40 \times 1$
L1: conv1_block	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$300 \times 40 \times 64$
L2: pool1	2×2, max pool, stride 2	$150 \times 20 \times 64$
L3: conv2_block	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$150 \times 20 \times 128$
L4: pool2	2×2, max pool, stride 2	$75 \times 10 \times 128$
L5: conv3_block	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$75 \times 10 \times 256$
L6: pool3	2×2, max pool, stride 2	$38 \times 5 \times 256$
L7: conv4_block	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$38 \times 5 \times 512$
L8: pool4	2×2, max pool, stride 2	$19 \times 3 \times 512$
L9: self-attention	$n_k = 4$	4×1536
L10: pool_time	avg pool	1536
L11: dense1	1536×256	256
L12: dense2	256×1251	1251

D. RESNETS

In addition, we devoted efforts to investigating the ResNets with the self-attention mechanism since the ResNets have gained great attention in computer vision problems, speech recognition, and speech emotion recognition [35], [37]. The key purpose of the ResNets is to solve the problem of

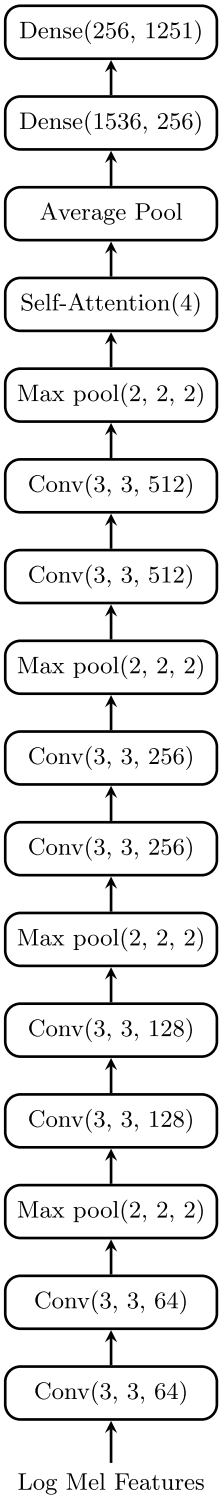

FIGURE 3. Modified VGG architecture with the self-attention layer and an average pool layer at the end for speaker identification. The ReLU and batch normalization layers are not shown.

performance degradation when there are a large number of hidden layers in a deep neural network. A common deep network generally is aimed at directly learning the underlying mapping. However, a deep ResNet is asked to fit a residual function [37]. The resulting residual mapping is more amenable to optimization because it is easier to push a residual to zero than to fit an underlying mapping [37]. In theory, given the target mapping $H(x)$ and the input of the first layer of the residual block x, the ResNet block fits the mapping as follows:

$$F(x) = H(x) - x, \tag{4}$$

and therefore, the original function becomes

$$H(x) = F(x) + x. \tag{5}$$

This can be accomplished by the addition of shortcut connections among the layers, as shown in Figure 4 [37]. As seen from Figure 4, the shortcut connections perform an identity mapping, and the inputs are added to the output of the multiple layers. Obviously, such a ResNet block is differentiable and can therefore be trained with traditional backpropagation.

Following the concept of deep residual learning, the ResNet, which was originally proposed for image object detection, is composed of a number of residual units (cf. Figure 4), where each residual unit consists of two convolutional layers with 3×3 filter sizes, batch normalization (BN) [47] is applied after each convolution, and ReLU activation functions are applied after the first convolution and after the shortcut connection addition operation.

TABLE 3. Modified ResNet-18 architecture with the self-attention layer and an average pool layer at the end for speaker identification. The ReLU and batch normalization layers are not shown. Each row specifies the number of convolutional filters and their sizes as filter_size × filter_size, # filters. Here, the self-attention layer corresponds to the layer introduced in Section III-B.

Layer	ResNet-18	Output ($T \times F \times C$)
Input	–	$300 \times 40 \times 1$
L1: conv1	$7 \times 7, 32$, stride 1	$300 \times 40 \times 32$
L2: pool1	3×3, max pool, stride 2	$150 \times 20 \times 32$
L3: conv2_block	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 2$	$75 \times 10 \times 32$
L4: conv3_block	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$38 \times 5 \times 64$
L5: conv4_block	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$19 \times 3 \times 128$
L6: conv5_block	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$10 \times 2 \times 256$
L7: self-attention	$n_k = 4$	4×512
L8: pool_time	avg pool	512
L9: dense1	512×256	256
L10: dense2	256×1251	1251

In this work, we modify the original **ResNet-18** with 18 hidden layers and extend it by adding the structured self-attention layer for speaker identification. The proposed ResNet-18 architecture is shown in Table 3.

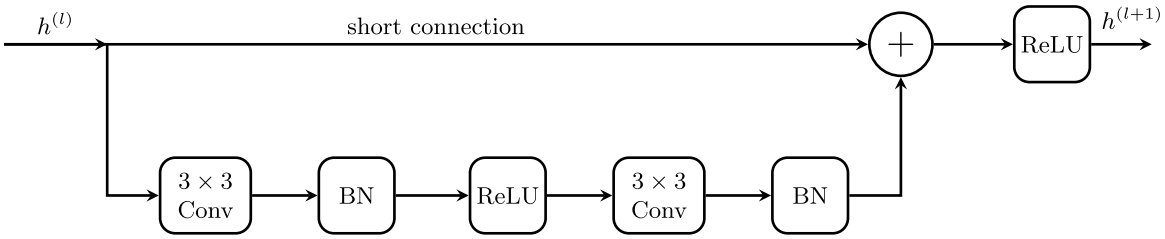

FIGURE 4. A basic residual unit in ResNets, which consists of convolutional layers (Conv) with 3 × 3 filters, batch normalization (BN) layers, and the ReLU activation function [37].

E. LOSS

The proposed network is asked to classify speakers using a multi-class cross entropy objective function, which is commonly used for image object detection and speaker recognition in neural networks [19], [20], [48]. Unlike the previously reported systems that were trained to predict speaker labels from frames [17], [29], [30], our system is trained to predict speakers from variable-length segments. Let us consider a dataset with N training examples from K speakers. Given a speech segment consisting of T input frames $x_1^{(n)}, x_2^{(n)}, \ldots, x_T^{(n)}$, let $p(y_k|x_{1:T}^{(n)})$ be the prediction probability of the deep network model for the k-th speaker. The cross-entropy objective function is formally defined as follows:

$$l = -\sum_{n=1}^{N}\sum_{k=1}^{K} d_{nk} \log(P(y_k|x_{1:T}^{(n)})), \tag{6}$$

where the quantity d_{nk} is 1 if the speaker label for the n-th training segment is k; otherwise, it is 0. In the training process, the parameters of a network are optimized towards minimizing the cross-entropy objective by the backpropagation algorithm [49]–[51]. Therefore, the network training algorithm is shown in Algorithm 1.

Algorithm 1 The Neural Network Training Optimization Algorithm

Require: Learning rate
Require: Initial parameters θ
1: **while** stopping criterion not met **do**
2: Sample a minibatch of m examples from the training set $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}$ with corresponding target labels $\mathbf{y}^i$.
3: Compute the loss based on the (6).
4: Compute gradient estimate: $\hat{g}$.
5: Apply update with the gradient estimate $\hat{g}$ for the parameters θ.
6: **end while**

IV. EXPERIMENTS

A. SELECTED DATA

We use the VoxCeleb database to evaluate the effectiveness of the proposed system [5], which is a large-scale text-independent speaker identification corpus including 153 486 utterances for 1,251 celebrities, extracted from videos uploaded to YouTube. As shown in Figure 5, the 153 486 utterances are of varying duration, ranging from 3.96 seconds to 144.92 seconds. The dataset is gender-balanced, with 55% of the speakers being male. The speakers span a wide range of different ethnicities, accents, professions, and ages. Moreover, there are a large number of challenging multi-speaker acoustic environments in the dataset, including red carpet, outdoor stadium, quiet studio interviews, speeches given to large audiences, excerpts from professionally shot multimedia, and videos shot on hand-held devices. As a consequence, all utterances are degraded by real-world noise, consisting of background chatter, laughter, overlapping speech, and room acoustics, and there is a range in the quality of the recording equipment and the channel noise. Figure 5 presents the distribution of the 153 486 utterances in the VoxCeleb database. The speaker identification task was introduced as shown in Table 4 [5]. In the following experiments, we follow the official split regarding the dataset and report the top-1 and top-5 accuracies.

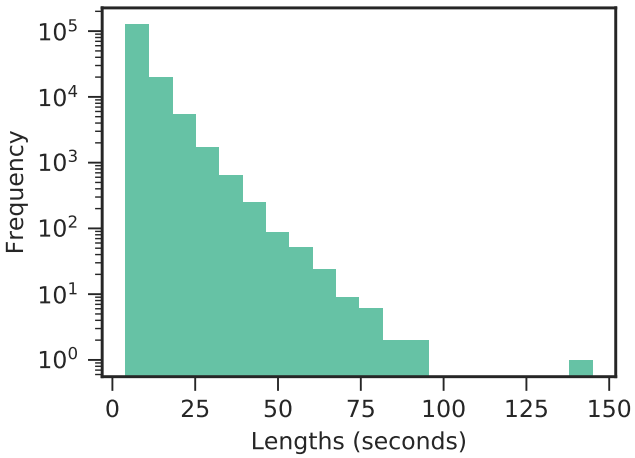

FIGURE 5. Distribution of the utterance lengths in the VoxCeleb database.

B. EXPERIMENTAL SETUP

As for the acoustic feature extraction, pre-emphasis with a factor of 0.97 is first conducted. Then, 40 dimensional log-mel filterbanks using a Hamming window with

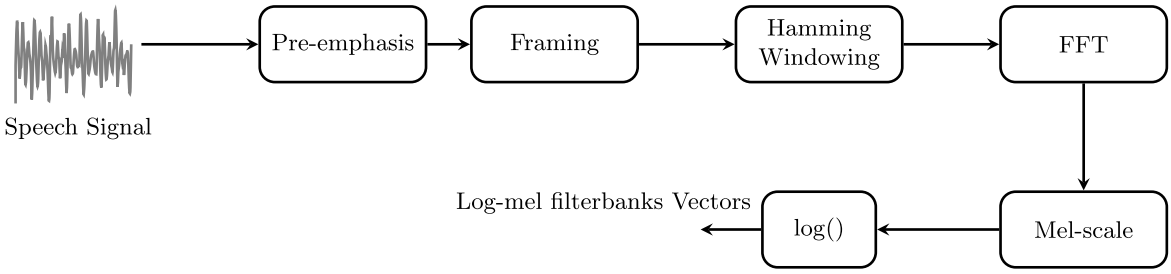

FIGURE 6. Computation process for log-mel filterbanks used as the inputs to the deep networks. 'FFT' corresponds the Fast Fourier Transform.

TABLE 4. Number of instances for the speaker identification task in VoxCeleb.

Train	Validation	Test	$\sum$
138 327	6 908	8 251	153 486

a frame-length of 25 ms and a frame-shift of 10 ms are extracted. Since the utterances in the VoxCeleb dataset are of varying duration (up to 144.92 s), we fix the length of the input sequence to 3 seconds. These end up as log-mel filterbank of size 40×300 for a 3-second utterance. Figure 6 presents the whole process to extract the log-mel filterbanks features. In addition, mean and variance normalization is performed on every frequency bin of the mels to obtain zero mean and unit variance, which plays a key role in this system as found in [5].

In the testing stage, all the testing utterances with different duration are tested on the same model. Since the duration is arbitrary, we feed the testing speech utterances to the trained neural network one by one.

C. IMPLEMENTATION DETAILS AND NETWORK TRAINING

All experiments in the paper are implemented by the widely used deep learning tool TensorFlow [52]. We set the batch size as 128 and train neural networks on one NVIDIA GTX 1080 Ti GPU. We use the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-9}$ [53]. The warmup process for varying the learning rate [46], which increases the learning rate linearly for the first predefined training steps and then decreases it proportionally, is adopted to speed up the learning process. Moreover, we use grid search to determine a suite of hyper-parameters such as weight decay, the size of the embedding layer, the dropout probability, and the maximum gradient norm. To reduce sensitivity to utterance length, it is desirable to train the network on speech chunks that capture the range of duration we expect to encounter at test time (e. g., a few seconds to a few minutes, as shown in Figure 5). However, due to GPU memory limitations, we have to find a tradeoff between the minibatch size and maximum training example length. Moreover, to remedy overfitting [5], we randomly sample 3-second segments from each utterance for the training and validation data in the training process, which can be treated as a simple data augmentation method.

D. METHODS FOR COMPARISON

We compare the following state-of-the-art methods to evaluate the effectiveness of the proposed approach.

- I-Vectors + SVM [5]: This approach was previously implemented along with the release of the VoxCeleb dataset [5]. The implementation of this system is explained as follows: the GMM-UBM system was first built by using 13-dimensional MFCCs as input. The cepstral mean and variance normalization (CMVN) is applied on the features. Using the conventional GMM-UBM framework, a single speaker-independent universal background model (UBM) of 1024 mixture components is trained for 10 iterations from the training data. gender-independent i-vector extractors [10] are trained on the VoxCeleb dataset to produce 400-dimensional i-vectors. Probabilistic LDA (PLDA) is then used to reduce the dimension of the i-vectors to 200. For identification, a one-vs-rest binary SVM classifier is trained for each speaker m ($m \in 1 \ldots K$). All feature inputs to the SVM are L2 normalized, and a held-out validation set is used to determine the C parameter (which determines the tradeoff between maximizing the margin and penalizing training errors). Classification during test time is performed by choosing the speaker corresponding to the highest SVM score.
- I-Vectors + PLDA + SVM [5]: This system is similar to the I-Vectors + SVM system, except that the PLDA score function is applied.
- I-Vectors + LogReg [6]: This system takes 60-dimensional MFCCs features with delta and double-delta coefficients to train a 2048-components full covariance GMM UBM model, resulting in a 600-dimensional i-vector. For the closed-set speaker identification, a multi-class LogReg is selected for the inference.
- VGG-like CNN + **TAP** [5]: This approach uses 512×300 dimension spectrograms for a fixed 3 second chunk as the inputs to a VGG-like CNN with proper modifications for the speaker identification task. A temporal average pooling (TAP) layer is used after the **fc6** layer, which makes the network invariant to the length of the input speech segment.
- ResNet34 + {TAP, SAP, LDE} [6]: This system corresponds to three recently reported speaker identification systems based on 34-layer ResNets (ResNet34) with

TAP, self-attention pooling (SAP) [54], and learnable dictionary encoding-based pooling (LDE) [6]. For the acoustic features, 64 dimensional Fbanks that are mean-normalized over a sliding window of up to 3 seconds are adopted as the inputs to ResNet34. Before training these deep nets, a frame-level energy-based voice activity detection (VAD) selects the features corresponding to voice frames.

TABLE 5. The results for speaker identification on VoxCeleb (higher is better).

Accuracy	Top-1 (%)	Top-5(%)
I-Vectors + SVM [5]	49.0	56.6
I-Vectors + PLDA + SVM [5]	60.8	75.6
I-Vectors + LogReg [6]	65.8	81.4
VGG-like CNN+ TAP [5]	80.5	92.1
ResNet-34 + TAP [6]	88.5	94.9
ResNet-34 + SAP [6]	89.2	94.1
ResNet-34 + LDE [6]	89.9	95.7
VGG-like CNN+Self-Attention (ours)	88.2	93.8
ResNet-18+Self-Attention (ours)	**90.8**	**96.5**

E. RESULTS FOR SPEAKER IDENTIFICATION

Table 5 presents the experimental results achieved by our proposed VGG-like CNN and ResNet-18, the traditional i-vector based methods, as well as the two recently proposed CNN-based methods on the VoxCeleb database. It can be obviously determined from the table that our proposed methods reach 88.2% and 90.8% of the top-1 accuracy and 93.8% and 96.5% of the top-5 accuracy for the VGG-like CNN + Self-Attention and ResNet-18 + Self-Attention approaches, respectively, which outperform the traditional i-vector-based methods in terms of top-1 and top-5 accuracies by a large margin, suggesting the effectiveness of deep neural networks for speaker identification over the traditional methods, such as the i-vector-based methods. Further, the VGG net and ResNets with the self-attention layer perform better than the VGG and ResNets alternatives, and it is worth noting that the ResNet-18 with the structured self-attention mechanism obtains the best accuracies. These findings indicate that the ResNets are more suited for speaker identification than the VGG nets. Most importantly, they strongly suggest the key role of the self-attention in the proposed VGG and ResNet-18 architectures.

F. EFFECT OF DIFFERENT ACOUSTIC FEATURES

Here, we start to investigate the effect of different acoustic features on the proposed VGG-like and ResNet CNNs. In addition to FBank acoustic features, two widely used features, spectrograms and MFCCs, which have been commonly adopted for speaker identification, are used for comparison here. Similar to the computation process shown in Figure 6, a total of 40 coefficients were extracted for MFCCs and FBanks, and 512 for spectrograms. All features were normalized to have zero-mean and zero-unit variance.

TABLE 6. The results of the three most frequently used acoustic features for speaker identification on VoxCeleb (higher is better). Here, Spectr. corresponds to the spectrograms feature.

Model	Feature Type	Top-1 (%)	Top-5(%)
VGG-like CNN [5]	Spectr.	80.5	92.1
VGG-like CNN+Self-Attention (ours)	Spectr.	85.3	92.9
ResNet-18+Self-Attention (ours)	Spectr.	87.2	93.3
VGG-like CNN+TAP [5]	MFCCs	82.4	92.8
VGG-like CNN+Self-Attention (ours)	MFCCs	87.4	93.5
ResNet-18+Self-Attention (ours)	MFCCs	88.5	94.8
ResNet-34 [6]	FBank	89.9	95.7
VGG-like CNN+Self-Attention (ours)	FBank	88.2	93.8
ResNet-18+Self-Attention (ours)	FBank	**90.8**	**96.5**

Table 6 shows the accuracies obtained using spectrograms, MFCCs, and FBank. First, we can easily observe that models using the FBank features always outperform the spectrograms and MFCCs features in our experiments, which echoes the finding reported in [35]. Second, when the input features are the same, due to the use of the structured self-attention mechanism, our proposed VGG and ResNet variants outperform the previously proposed VGG and ResNet architectures on the benchmark database. In the following sections, we place our focus only on the FBank features.

G. IMPACT OF TEMPORAL POOLING LAYERS

As mentioned in Section I, it is common to apply a temporal pooling layer after stacked layers in a deep net to result in a fixed-length embedding vector. Generally, there are three types of temporal pooling layers, i. e., average pooling, maximum pooling, and the combination of average pooling and standard deviation pooling. As shown in the diagram of the proposed system (cf. Figure 1), an average pooling layer is inserted between the self-attention layer and the last fully connected layer. Here, we perform a systematic investigation of the impact of the three types of common temporal pooling layers on the proposed VGG and ResNet-18.

Table 7 shows the results for different temporal pooling layers on the VoxCeleb database. As can be shown in the table, the proposed networks fluctuate according to the various temporal pooling layers. However, the average pooling always boosts the performance of the proposed networks, which echoes the consistent findings reported in [5].

H. SPEAKER CLUSTERING

Finally, based on our proposed ResNet18 speaker identification network, we present, here, that it is possible to cluster an unknown speaker through the activations of an upper (dense or softmax) layer of a pre-trained identification CNN as a feature vector. Figure 7 visualizes the individual output vectors produced by the snippets from 10 unknown speakers (i. e., never encountered during the

TABLE 7. Results of the speaker identification experiment on VoxCeleb for different common temporal pooling layers when the Fbank features are extracted for the experiments. (higher is better).

Model	Pooling	Top-1 (%)	Top-5(%)
VGG-like CNN [5]	Average Pooling	82.4	92.8
ResNet-34 [6]	Average Pooling	88.5	94.9
VGG-like CNN+Self-Attention (ours)	Maximum Pooling	86.9	93.4
VGG-like CNN+Self-Attention (ours)	Average Pooling	88.2	93.8
VGG-like CNN+Self-Attention (ours)	Average + Std Pooling	87.5	93.5
ResNet-18+Self-Attention (ours)	Maximum Pooling	88.1	94.4
ResNet-18+Self-Attention (ours)	Average Pooling	90.8	96.5
ResNet-18+Self-Attention (ours)	Average + Std Pooling	89.8	95.5

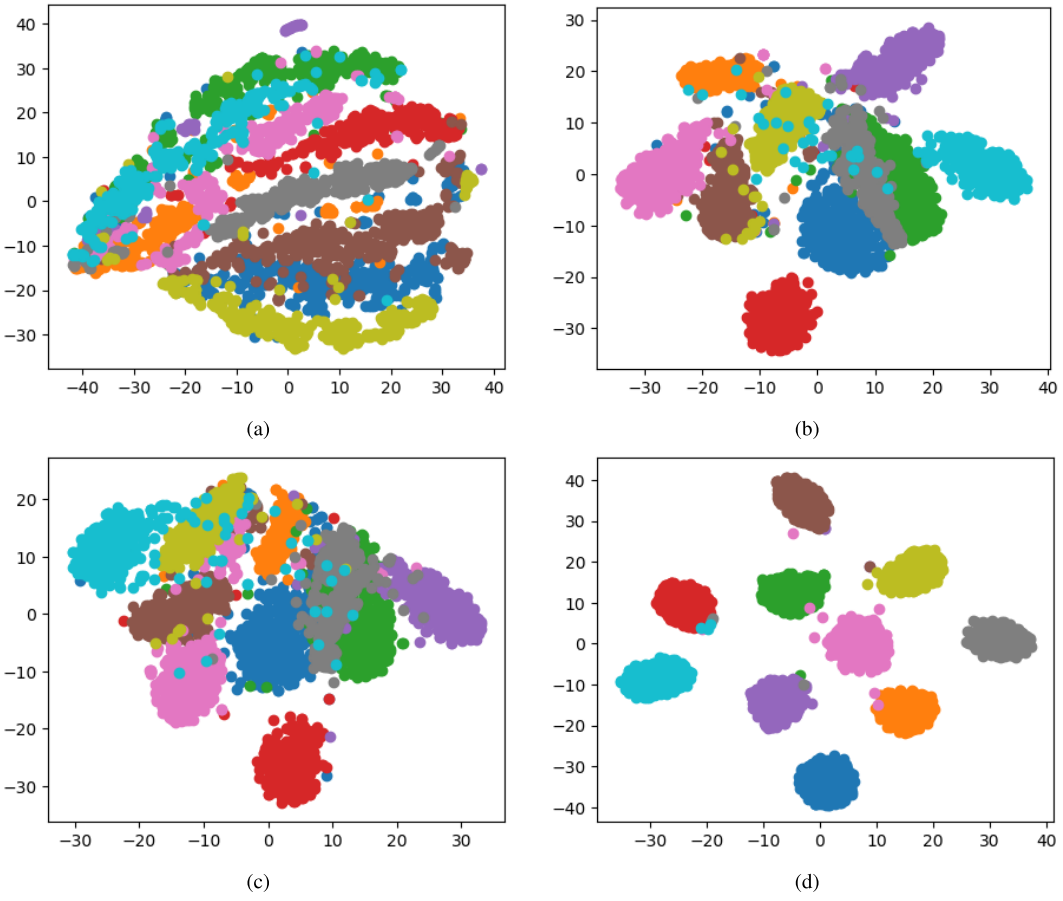

FIGURE 7. Speaker Embedding visualization using t-SNE on the basis of the FBanks, output vectors of the average pooling layer, softmax layer, and embedding layer (i. e., the fully connected before the softmax layer) for 10 speakers randomly selected from the VCTK database. Different colors represent embeddings from different speakers. (a) Fbanks. (b) Average pooling layer (i. e., L8 in Table 3). (c) Softmax layer (i. e., L10 in Table 3). (d) Embedding layer (i. e., L9 in Table 3).

original identification-targeted training) randomly chosen from the VCTK database [55] for the average pooling layer (the L8 layer in Table 3), the embedding layer (the L9 layer in Table 3) and the softmax layer (the L10 in Table 3) in the ResNet18 network trained to recognize 1251 speakers, using the popular visualization method t-SNE [56] with the cosine metric. Although there is a mismatch between the VoxCeleb database used to train the network and the VCTK database, we observe a very clear separation among different speakers for the resulting vectors from the L9 layer (cf. Table 3), which suggests that the low-dimensional speaker embeddings from the L9 layer can be treated as a useful latent space for speaker separation. Moreover, the network can efficiently learn higher-level representations of the low-level acoustic features when compared with the FBanks.

V. CONCLUSIONS

The research that related to speech processing has currently attracted increasing attention. It has motivated application that have been working effectively in many fields, including the most recent ones: banking, smart homes, smarts cities and the Internet of vehicles. This work proposes novel methods for text-independent speaker identification, where two representative convolutional neural networks, VGG and ResNet, are extended by a structured self-attention mechanism, which can aggregate relevant information from different locations of the variable-length input utterance. The proposed networks are extensively evaluated on the large VoxCeleb database. We observe that the proposed methods with the self-attention mechanism outperform the traditional i-vector-based methods and other recently proposed deep convolutional networks.

In conclusion, our contributions are summarized as follows:

(1) We propose a novel approach of exploiting the structured self-attention layer with multiple attention hops which learns rich speaker characteristics from different aspects of the input sequence. It turns out that these exploited speaker characteristics can be used to improve speaker identification accuracy.
(2) We add the self-attention layer to two representative CNNs, ie, VGG and ResNets, to build speaker identification models.
(3) Through experimental results, we demonstrate that our proposed methods outperform the traditional methods and other deep learning methods by a large margin. Our best model improves on previous state-of-the-art performance on the VoxCeleb speech identification task by 0.9 %.

In the future, we plan to investigate the effectiveness of the proposed networks for other related applications, such as speaker verification, speaker diarization, and speech emotion recognition.

REFERENCES

[1] J. P. Campbell, Jr., "Speaker recognition: A tutorial," *Proc. IEEE*, vol. 85, no. 9, pp. 1437–1462, Sep. 1997.
[2] R. Togneri and D. Pullella, "An overview of speaker identification: Accuracy and robustness issues," *IEEE Circuits Syst. Mag.*, vol. 11, no. 2, pp. 23–61, 2nd Quart., 2011.
[3] J. H. L. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 74–99, Nov. 2015.
[4] E. Kiktova and J. Juhar, "Speaker recognition for surveillance application," *J. Elect. Electron. Eng.*, vol. 8, no. 2, pp. 19–22, 2015.
[5] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. INTERSPEECH*, 2017, pp. 1–6.
[6] 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. Speaker Odyssey*, 2018, pp. 74–81.
[7] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Proc. Interspeech*, 2017, pp. 1487–1491.
[8] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 4930–4934.
[9] S. S. Tirumala, S. R. Shahamiri, A. S. Garhwal, and R. Wang, "Speaker identification features extraction methods: A systematic review," *Expert Syst. Appl.*, vol. 90, pp. 250–271, 2017.
[10] R. Sarikaya, B. L. Pellom, and J. H. Hansen, "Wavelet packet transform features with application to speaker identification," in *Proc. 3rd IEEE Nordic Signal Process. Symp.*, 1998, pp. 81–84.
[11] D. A. Reynolds and D. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 1, pp. 72–83, Jan. 1995.
[12] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, "On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition," *Bell System Tech. J., The*, vol. 62, no. 4, pp. 1075–1105, Apr. 1983.
[13] F. K. Soong, A. E. Rosenberg, B.-H. Juang, and L. R. Rabiner, "Report: A vector quantization approach to speaker recognition," *AT&T Tech. J.*, vol. 66, no. 2, pp. 14–26, Mar./Apr. 1987.
[14] M. N. Schmidt, J. Larsen, and F.-T. Hsiao, "Wind noise reduction using non-negative sparse coding," in *Proc. IEEE Workshop Mach. Learn. Signal Process.*, Aug. 2007, pp. 431–436.
[15] N. Dehak, P. J. Kenny, R. Dehak, D. Pierre, and O. Pierre, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 4, pp. 788–798, May 2011.
[16] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Comput. Speech Lang.*, vol. 20, nos. 2–3, pp. 210–229, 2006.
[17] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, vol. 14, May 2014, pp. 4052–4056.
[18] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2017, pp. 999–1003.
[19] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*, Apr. 2018, pp. 5329–5333.
[20] M. McLaren, D. Castan, M. K. Nandwana, L. Ferrer, and E. Yılmaz, "How to train your speaker embeddings extractor," in *Proc. Odyssey, Speaker Lang. Recognit. Workshop*, Les Sables-d'Olonne, France, 2018, pp. 1–9.
[21] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 308–311, May 2006.
[22] P. Kenny and P. Dumouchel, "Experiments in speaker verification using factor analysis likelihood ratios," in *Proc. ODYSSEY-Speaker Lang. Recognit. Workshop*, 2004, pp. 219–226.
[23] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Amsterdam, The Netherlands: Elsevier, 2013.
[24] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE 11th Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.
[25] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. INTERSPEECH*, 2011, pp. 249–252.
[26] S. O. Sadjadi, S. Ganapathy, and J. W. Pelecanos. (2016). "The IBM 2016 speaker recognition system." [Online]. Available: https://arxiv.org/abs/1602.07291
[27] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 1695–1699.
[28] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 5115–5119.
[29] Y. Konig, L. Heck, M. Weintraub, and K. Sonmez, "Nonlinear discriminant feature extraction for robust text-independent speaker recognition," in *Proc. RLA2C, ESCA Workshop Speaker Recognit. Commercial Forensic Appl.*, 1998, pp. 72–75.
[30] L. P. Heck, Y. Konig, M. K. Sönmez, and M. Weintraub, "Robustness to telephone handset distortion in speaker recognition by discriminative feature design," *Speech Commun.*, vol. 31, nos. 2–3, pp. 181–192, 2000.
[31] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2016, pp. 165–170.

[32] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for LVCSR," in *Proc. ICASSP*, Shanghai, China, Mar. 2016, pp. 4955–4959.
[33] G. Saon, T. Sercu, S. Rennie, and H.-K. J. Kuo, "The IBM 2016 English conversational telephone speech recognition system," in *Proc. INTERSPEECH*, San Francisco, CA, USA, 2016, pp. 7–11.
[34] S. Hershey *et al.*, "CNN architectures for large-scale audio classification," in *Proc. ICASSP*, New Orleans, LA, USA, 2017, pp. 131–135.
[35] J. Deng, F. Eyben, B. Schuller, and F. Burkhardt, "Deep neural networks for anger detection from real life speech data," in *Proc. 7th Int. Conf. Affect. Comput. Intell. Interact. Workshops Demos (ACIIW)*, Oct. 2017, pp. 1–6.
[36] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[38] C. Szegedy *et al.* (2015). "Going deeper with convolutions." [Online]. Available: https://arxiv.org/abs/1409.4842
[39] Z. Lin *et al.* (2017). "A structured self-attentive sentence embedding." [Online]. Available: https://arxiv.org/abs/1703.03130
[40] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 577–585.
[41] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 4945–4949.
[42] D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur, "A time-restricted self-attention layer for ASR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5874–5878.
[43] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2227–2231.
[44] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio. (2014). "End-to-end continuous speech recognition using attention-based recurrent NN: First results." [Online]. Available: https://arxiv.org/abs/1412.1602
[45] D. Bahdanau, K. Cho, and Y. Bengio. (2014). "Neural machine translation by jointly learning to align and translate." [Online]. Available: https://arxiv.org/abs/1409.0473
[46] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[47] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167
[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[49] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.
[50] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. M "uller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*. London, U.K.: Springer-Verlag, 1998, pp. 9–50. [Online]. Available: http://dl.acm.org/citation.cfm?id=645754.668382
[51] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.
[52] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. OSDI*, Berkeley, CA, USA, 2016, pp. 265–283.
[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, Dec. 2014.
[54] G. Bhattacharya, J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification," in *Proc. INTERSPEECH*, 2017, pp. 1517–1521.
[55] C. Veaux *et al.*, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," Tech. Rep., 2016. doi: 10.7488/ds/1994.
[56] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

NGUYEN NANG AN received the B.S. degree in information technology and the M.S. degree in computer science from Hanoi Pedagogical University 2, Vietnam, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree in computer science with the Chongqing University of Posts and Telecommunications. His research interests include computer science, artificial intelligence, and information security safeguards.

NGUYEN QUANG THANH was born in Me Linh, Hanoi, Vietnam, in 1988. He received the B.S. degree in information technology and the M.S. degree in computer science from Hanoi Pedagogical University 2, Vietnam, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree in computer science with the Chongqing University of Posts and Telecommunications. His research interests include computer science, artificial intelligence, and data mining.

YANBING LIU received the M.S. degree in computer science and technology from the Beijing University of Posts and Telecommunications, China, in 2001, and the Ph.D. degree from the University of Electronic Science and Technology, China, in 2007.

He is currently an Executive Director of the Chongqing Youth Federation of Science and Technology. He is currently a Professor and a Ph.D. Supervisor with the Chongqing University of Posts and Telecommunications. He has authored over 60 refereed papers. His research interests include information security and management, security in cloud computing, and the Internet of Things. He was a recipient of the National Science and Technology Award and several Chongqing Science and Technology Awards.