Speaker Verification Using Multi-Scale K-Neighboring Residual network for Robust Embedding Learning

No Author Given

No Institute Given

Abstract. We propose a novel convolutional neural network (CNN) architecture for speaker verification, designed to effectively capture sequential and local interactions within input speech mel-filterbanks. Our approach integrates a series of multi-scale K-neighboring residual convolutional (MKRC) blocks, which enable each sub-feature to integrate contextual information from its local neighbors. This architecture facilitates the generation of speaker embeddings with an enhanced ability to differentiate between similar speakers. At first, Mel-filterbanks are extracted from each input speech, which is fed to the proposed model for speaker embedding generation. Euclidean distance is calculated between pairwise embeddings for performance evaluation. The proposed model has been compared with state-of-the-art methods using VoxCeleb1 dataset. Experimental results show that our model achives promising results in terms of Equal Error Rate (EER) and minimum Detection Cost Function (minDCF).

Keywords: Speaker Verification · Multi-scale K-Neighbor · Speaker Embedding.

1 Introduction

Speaker verification (SV) is the process of verifying whether two utterances belong to the same speaker or not. It analyzes pitch, tone, speaking style, and pronunciation, among other aspects of speech. This technology is applied in many different fields, including support of forensic investigations, access control mechanisms, validation of voice-activated devices, and security system enhancement. Speaker verification authenticates a claimed identity (1:1 match), whereas speaker identification [17] aims to recognize an unknown speaker from a known group (a 1:N match).

Due to significant advancements in speech signal processing and deep learning, speaker verification has undergone substantial transformations over the years. Most of the conventional speaker recognition methods were based on statistical models and binary hypothesis tests, such as Gaussian Mixture Models (GMMs)[16] and Hidden Markov Models (HMMs)[7]. With the advancement

of deep learning, more advanced architectures have emerged such as Time-Delay Neural Networks (TDNNs)[20], Residual Networks (ResNets)[6], ECAPA-TDNN[5] and Dense-Residual Networks. Such architectures can deeply represent the dissimilarities of time–frequency properties among different speakers, thereby enhancing the accuracy and robustness of speaker verification systems.

In [13], authors proposed the fusion of DenseNets and ResNets techniques to improve embedding learning . Their work consisted of building Dense-Residual (DenseR) blocks by combining dense connections with residual learning to collect complementing information without increasing model complexity compared to stacking more layers. Especially in view of model complexity, these blocks (implemented in sequential and parallel configurations) show better performance than standalone ResNets[6] or DenseNets [20]. The success of Dense-Residual networks emphasizes the possibilities of combining architectural strengths to extract richer and more discriminative speaker embeddings [13].

However, such methods process the entire sequence as a whole, by extracting temporal and spectral features at a single scale. This process can miss important local spatial relationships and multi-scale features within the audio signal, which may contain useful and relevant speaker-specific information. Inspired from the work presented in [12], we address a novel architecture using multi-scale k-neighboring residual convolutional (MKRC) blocks designed to process the input speech signal as a multi-scale feature. This is achieved by dividing the sequence into non-overlapping sub-features allowing each one of them to obtain information from its neighboring sub-features. This method enables our model to acquire spatially correlated information at multi resolution levels, which leads to a more accurate speaker verification.

Our study examines two key aspects of the MKRC architecture's performance: first, how different k-neighboring scales (k-values) affect feature learning, and second, how the number of MKRC blocks impacts the model's capacity to capture both local and spatial dependencies. Section 2 provides an overview of the proposed model design and the MKRC block architecture. Experimental results show that our model acts robusltly compared to the state-of-the-art baseline methods in terms of EER and minDCF [11].

2 Method

Our model is proposed to learn speaker embeddings that represent the unique vocal characteristics of different speakers. The overall architecture is composed of four main stages, namely the initial convolutional feature extractor, the stack of Multi-scale K-neighboring residual convolutional blocks, the attentive statistical pooling layer and the feed-forward network for embedding projection, as clearly illustrated in figure 1. We discuss the details and configurations of the proposed model in the following subsections.

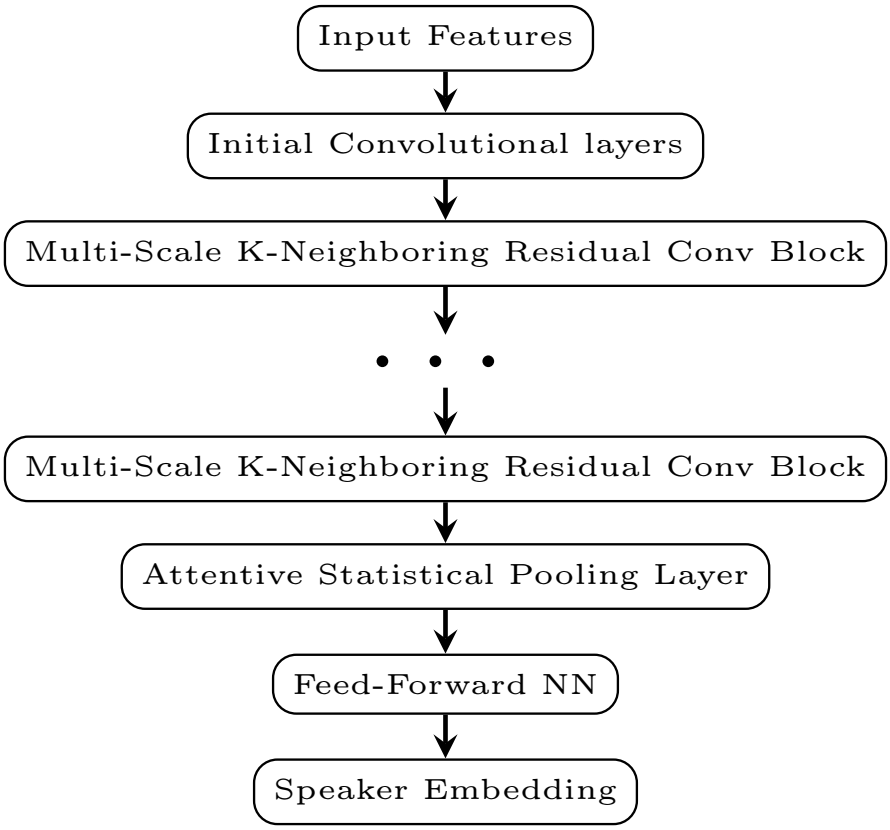

Fig. 1: The overall architecture of the proposed speaker verification model

2.1 Initial Convolutional Layers

The input Mel-filterbanks (shape: $112 \times T$) first pass through two 1D dilated convolutional layers, which serve as the initial feature extractors for the subsequent MKRC blocks, followed by a batch normalization layer and a ReLU activation function. We used dilated convolution over standard convolution because dilated convolution can capture longer time-frequency contextual information by expanding the receptive field without increasing kernel size or computational cost[12].

For 1D dilated convolution applied to a 2D input tensor $X \in R^{C*T}$ (Where C is the number of channels and T is the number of frames), the output is defined by:

$$Out(k,q) = \sum_{c} \sum_{t+d*s=q} In(c,t) * Weight(k,c,s) \tag{1}$$

Where d is a dilation rate, s is the filter width and $k = 1, ..., K$ where K is the total number of filters [3].

2.2 Multi-Scale K-Neighboring Residual Blocks

Figure 2 illustrates the workflow of the proposed MKRC block. The main contribution of the proposed work consists of using a stack of Multi-Scale K-Neighboring Residual Convolutional (MKRC) blocks. At first, the MKRC blocks divide the input feature map into non-overlapping sub-features along the frequency axis. This division allows the model to focus on smaller areas of the speech signal, which represent some different aspects of the audio signal, at multiple scales. In turn, each sub-feature is processed with a distinct convolutional kernel, allowing the model to extract diverse patterns across several frequency regions. The multi-scale structures can learn more detailed local spatial information compared to single-scale structures [8].

To improve feature learning, each sub-feature learns from its K neighboring sub-features using a sequential dependency mechanism. The model processes each sub-feature sequentially while allowing it to recover information from its neighbors. This process helps the model to capture both local spatial relationships (within individual sub-features) and global context (across neighboring sub-features), both of which are crucial for distinguishing speaker-specific characteristics, and learning dependencies between these sub-features. In fact, both global context and local spatial dependencies lead to better discrimination between different speakers.

The output of the dilated convolution for the sub-feature S_i is expressed as:

$$\widehat{S}_i = Conv(S_i + \sum_{k=1}^{K} \widehat{S}_{i-k}) \tag{2}$$

$K = 2$ is taken as an example, as shown in the figure 2. In equation 2, $\widehat{S}_{i-k}$ represents the output of sub-feature S_{i-k} after an element-wise summation is applied with its neighboring sub-features. The summation of these neighboring sub-features allows the model to combine both local and global information, which is helpful for a robust speaker embedding.

Following this aggregation step, the processed sub-features are concatenated, and the output feature is element-wise summed with the input feature map. This summation is known as a residual connection (also known as skip-connection), which is a technique that makes the training of deep learning models easier, especially in deeper networks [6].

To enhance the model's representational capacity, we implemented a sequence of multiple MKRC blocks. This sequence allows the network to capture more features at many levels of detail. Additionally, we can easily adjust the number of stacked blocks based on the dataset availability.

We can also adjust the number k of neighbors that individually interact with each sub-feature. This flexibility allows us to fine-tune the model's ability to balance between localization (by limiting the number of neighbors) and contextual learning (by considering more neighbors).

In the proposed approach, the model is designed to capture the most relevant characteristics of speakers voices. By combining the multi-scale feature processing and the K-neighbor mechanism, the MKRC architecture improves

the discriminative power of the speaker embeddings, contributing to more accurate and robust speaker verification systems.

The impact of different k values on the model performance and the number of stacked MKRC blocks, in terms of both EER and minDCF, will be evaluated in later sections.

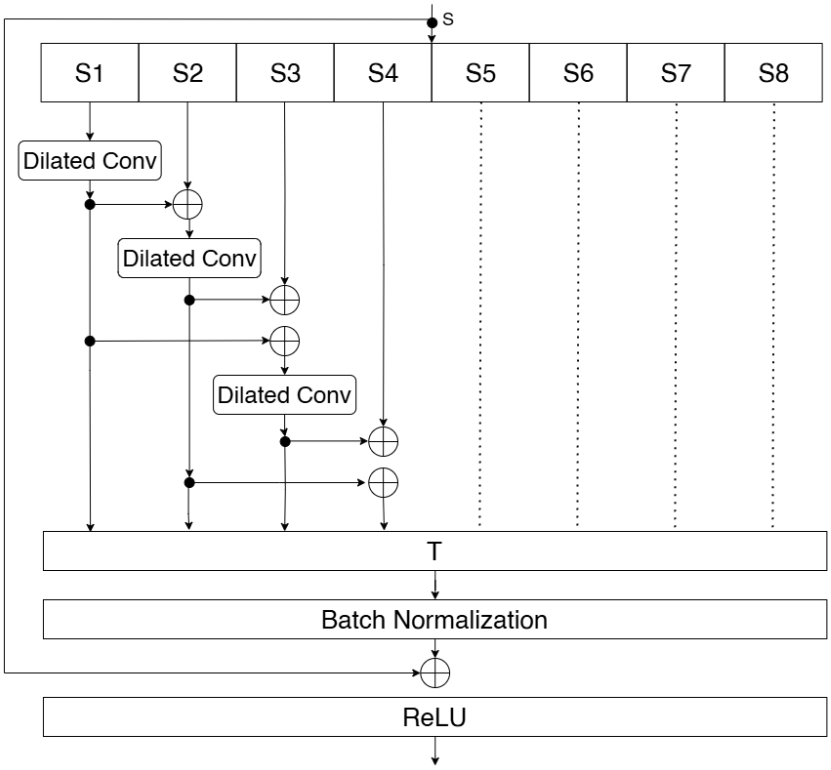

Fig. 2: Workflow of the MKRC block

2.3 Attentive Statistical Pooling Layer

The attentive statistic pooling (Att-SP)[15] layer dynamically aggregates temporal information by computing weighted mean and standard deviation across the time axis. Unlike uniform temporal averaging which treats all frames as equal, Att-SP learns to emphasize speaker-discriminative frames (such as stable vowels or salient consonants) that are more informative for speaker verification, through an attention mechanism. For each time step t, the attention weight α_t can be defined as:

$$\alpha_t = softmax(W * h_t + b) \tag{3}$$

where h_t is the frame-level feature, and W, b are the model's learnable parameters.

2.4 Fully Connected Embedding Layer

To extract speaker embedding vectors, we implemented two dense layers that transform the pooled vectors into lower-dimensional space. Batch normalization is applied after each dense layer to improve model's convergence. The first layer processes the pooled vectors to extract more features, and the final layer projects them to the embedding space.

During the training process, The embeddings are optimized using an appropriate loss function (AAM-softmax) [18] which adds an angular margin between speaker classes. This procedure enables the model to enhance inter-speaker variability while reducing intra-speaker differences in the embedding space.

For evaluation scoring, pairwise embeddings similarity scores are computed using the Euclidean distance [4]. Given two speaker embeddings x and y extracted from two utterances respectively, their dissimilarity score $d(x, y)$ is calculated as:

$$d(x, y) = \|x - y\|_2 = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{4}$$

Where $\|.\|_2$ is the L2 norm, also known as Euclidean norm and n is the dimension of the embedding vector. This distance measure serves as our primary verification score, where lower values indicate higher speaker similarity.

Noting that, the ability to capture local and global information is achieved by enabling the interaction between neighboring sub-features across multiple scales. Unlike previously published techniques, where features are separately processed [6] [20] [5], our model learns relationships between fine-grained spectral details and broader vocal traits through proposed K-neighboring information exchange. This leads the model to generate robust and discriminative speaker embeddings.

3 Experimental Setup

3.1 Dataset

The used dataset is VoxCeleb1[14] dataset. For the training set, VoxCeleb1 contains over 100,000 utterances from 1,211 speakers, collected from YouTube videos under real-world, unconstrained conditions. The dataset is gender balanced, with 55% of male speakers. To evaluate our model, we used VoxCeleb1 test set, which contains over 6,000 utterances from 40 speakers, with no overlapping identity between training and test sets.

3.2 Input features

For our speaker verification task, Mel-filterbanks were extracted from each utterance and used as input features to the MKRC network.

The procedure of extracting mel-filterbanks is described as follows:

First, we apply a pre-emphasis filter on each speech signal with a coefficient set to 0.97. Next, we divide the signal into overlapping frames with a 25ms frame length and a 10ms frame shift. A custom window function named Povey [2] is applied on each overlapped frame.

Later on, The Short-Time Fourier Transform (STFT) process is performed on each windowed frame to obtain the power spectrum of the signal. To produce mel-filterbanks, a set of 112 triangular filters (111 Mel filters and an energy dimension) was applied on the resulted power spectrum, spaced on the Mel scale. This scale provides a representation that conforms better to human auditory system on the frequency resolution. The transformation of frequency f in Hz to the mel scale m can be described in the equation (5):

$$m = 2595 * log_{10}(1 + \frac{f}{700}) \tag{5}$$

The triangular filterbanks smooth the energy distribution across close frequency bands, which results of a unique, information-rich design that is suitable as an input for our speaker verification system. The mel-filterbanks can be visualized in figure 3.

The resulted filterbanks are of dimension 112 frequency bands $\times$ 300 time frames, which will be considered as inputs of the proposed model.

Attentive Statistical Pooling (ASP)[15] is used to generate utterance-level embeddings, without any data augmentation.

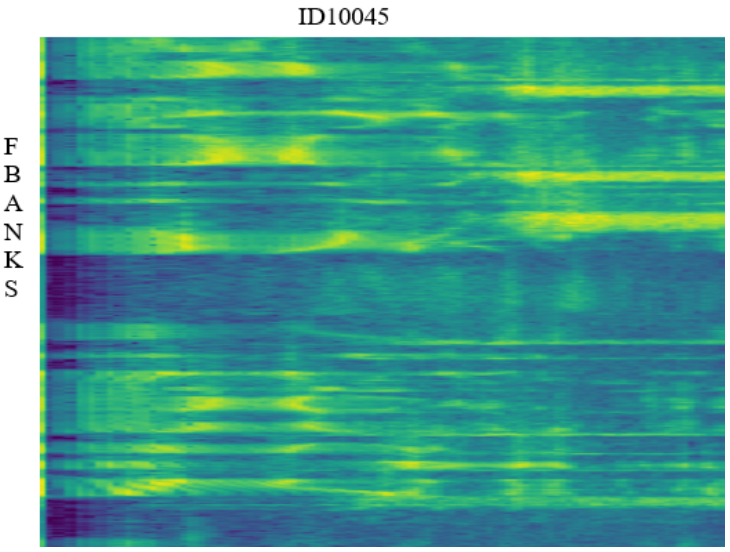

Fig. 3: Mel-filterbank features

3.3 Training settings

The proposed model was implemented using the PyTorch 2.5.1 framework[1] and trained on the ROCm 6.2 platform running Ubuntu 22.04 LTS [10], leveraging AMD's HIP backend for optimized GPU acceleration.

For both training and testing, a fixed 3-second time segment is randomly extracted from each utterance. The model is trained using Additive Margin Softmax (AAM-softmax) loss using Adam optimizer [21]. Parameters of the proposed model are presented in table 1.

Table 1: Settings of the proposed model.

Type	Details
Input features	Number of frames : 300
	Number of frequency bins : 112
Initial convolutional layers	Kernel size : 5
	Output channels : 512
	Dilation rate : 1,2
	Padding rate : 2,4
MKRC	Subfeature dimension : 64
	Kernel size : 5
	Padding and dilation rate : 2
	Number of blocks : 3
	Output channels in each block : 512
ASP	Number of channels : 128
	Output channels : 1024
FFNN	Neurons of first dense layer : 256
	Embedding dimension : 512
AAM Sofmax	Margin : 0.3
	Scaling factor : 30
Optimizer	Learning rate : 0.0005
	Degradation rate : 25% every 2 epochs
	Weight decay : 5×10^{-5}

System performance is evaluated via Equal Error Rate (EER) and minimum Detection Cost Function (minDCF). The detailed definitions of both EER and minDCF are referred to [11], with $P_{target} = 0.05$.

4 Results

4.1 Impact of K-Neighboring Scales number

In this subsection, we evaluate how the number of neighboring scales (K) affects verification performance using equal error rate (EER) and minimum detection cost (minDCF). For computational efficiency, batch size will be set to 512. The lowest scores are obtained at K=4 as shown in Table 2, this result indicatesthat

increasing K significantly improve the model performances. This 2.1% EER reduction from K=2 to K=4 indicates that the MKRC blocks are able to capture more global contextual information, resulting in more discriminative speaker embeddings. Based on the results, we will use K=4 for later experiments.

Table 2: Impacts of the K-Neighboring Scales number on the performance of the proposed model. The performance metrics are EER (in minDCF.

No. of scales	EER(%)	MinDCF
2	8.8971	0.49422
3	8.7858	0.48107
4	**8.7540**	**0.47248**

4.2 Impact of MKRC block number

Let us now evaluate how the number of MKRC blocks (N) affects verification performance using equal error rate (EER) and minimum detection cost (minDCF). As shown in Table 3, increasing the MKRC block number from N=3 to N=8 results in progressive performance improvements, with the best results achieved at N=8 (EER=8.66%, minDCF=0.469).However, at N=10, we can see a small regression in EER (8.85%). This might indicate that the model is starting to overfit the data. The 1.0% relative EER reduction from N=3 to N=8 indicates that deeper architectures model speaker characteristics better. Taking into account these results, we set N=8 for the next experiment.

Table 3: Impacts of the MKRC blocks number on the performance of the proposed model. The performance metrics are EER (in %) and minDCF.

No. of blocks	EER(%)	MinDCF
3	8.7540	0.47248
4	8.7723	0.46262
6	8.6903	0.47163
8	**8.6638**	**0.46925**
10	8.8494	0.48224

4.3 Comparison to different methods

For fair comparison against existing methods, a batch size of 64 was used in this experiment. Table 4 compares our model performance with state-of-the-art methods such as RawNet3, ResNet-SE, TDNN and ECAPA-TDNN [9] [6] [20] [5] [19].

As shown in Table 4, our model achieved EER=7.87% and minDCF=0.447. Results demonstrate that our model slightly outperforms RawNet3 (EER=8.71%), ResNet-SE (EER=11.52%) and conventional TDNN (8.51% EER), with narrowing the gap with ECAPA-TDNN (6.44% EER). MKRC shows a 7.5% improvement in minDCF compared to standard TDNN (0.447 vs 0.462), indicating improved reliability for real-world applications. Although ECAPA-TDNN is the best performer, our method exhibits a competitive performances compared to baseline speaker verification methods, validating MKRC's ability to learn discriminative speaker representations based on its unique multi-scale approach.

The achieved results validate MKRC's ability to learn discriminative speaker representations based on its unique k-neighboring approach. By integrating the combination of both local and global information, the model is able to generate robust speaker embeddings that are very useful in the context of speaker verification tasks.

Table 4: Scores of EER and minDCF obtained by our method compared to different methods.

Methods	EER(%)	MinDCF
RawNet3	8.71	0.4882
ResNet-SE	11.52	0.5675
TDNN	8.51	0.4624
ECAPA-TDNN	6.44	0.3784
MKRC	**7.8685**	**0.44745**

4.4 Conclusion

This work introduces a new architecture for speaker verification that processes input features in a multi-resolution manner, where each sub-feature dynamically incorporates information from its surrounding elements under a multi-scale perspective. Our model gathers both fine-grained local dependencies and global speaker characteristics by considering spectral and temporal patterns as hierarchical components. The main contribution lies in letting each sub-feature accumulate context from its K-nearest neighbors.

Through detailed experiments, we found that expanding the number of neighbouring scales (K) and MKRC blocks (N) consistently boosts performance. The best results were obtained at K=4 and N=8 blocks, achieving an EER of 7.87% and a minDCF of 0.447. Our model shows a competitive results compared to multiple state-of-the-art methods, including standard TDNN, RawNet3 and ResNet-SE.

References

1. Ahmed, K., Li, T., Ton, T., Guo, Q., Chang, K.W., Kordjamshidi, P., Srikumar, V., Van den Broeck, G., Singh, S.: Pylon: A pytorch framework for learning with constraints. In: NeurIPS 2021 Competitions and Demonstrations Track. pp. 319–324. PMLR (2022)
2. Boulianne, D.P.A.G.G., Goel, L.B.O.G.N., Qian, M.H.P.M.Y., Stemmer, P.S.J.S.G., Veselyd, K.: Trope (2012)
3. Chaudhary, N., Misra, S., Kalamkar, D., Heinecke, A., Georganas, E., Ziv, B., Adelman, M., Kaul, B.: Efficient and generic 1d dilated convolution layer for deep learning. CoRR **abs/2104.08002** (2021), https://arxiv.org/abs/2104.08002
4. Danielsson, P.E.: Euclidean distance mapping. Computer Graphics and image processing **14**(3), 227–248 (1980)
5. Desplanques, B., Thienpondt, J., Demuynck, K.: Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. arXiv preprint arXiv:2005.07143 (2020)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
7. Ilyas, M.Z., Samad, S.A., Hussain, A., Ishak, K.A.: Speaker verification using vector quantization and hidden markov model. In: 2007 5th Student Conference on Research and Development. pp. 1–5. IEEE (2007)
8. Jiang, S., Min, W., Liu, L., Luo, Z.: Multi-scale multi-view deep feature aggregation for food recognition. IEEE Transactions on Image Processing **29**, 265–276 (2019)
9. Jung, J.w., Heo, H.S., Kim, J.h., Shim, H.j., Yu, H.J.: Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification. arXiv preprint arXiv:1904.08104 (2019)
10. Kuznetsov, E., Stegailov, V.: Porting cuda-based molecular dynamics algorithms to amd rocm platform using hip framework: performance analysis. In: Supercomputing: 5th Russian Supercomputing Days, RuSCDays 2019, Moscow, Russia, September 23–24, 2019, Revised Selected Papers 5. pp. 121–130. Springer (2019)
11. Larcher, A., Lee, K.A., Ma, B., Li, H.: Text-dependent speaker verification: Classifiers, databases and rsr2015. Speech Communication **60**, 56–77 (2014)
12. Li, Y., Jiang, Z., Cao, W., Huang, Q.: Speaker verification using attentive multi-scale convolutional recurrent network. Applied Soft Computing **126**, 109291 (2022)
13. Liu, Y., Song, Y., McLoughlin, I., Liu, L., Dai, L.r.: An effective deep embedding learning method based on dense-residual networks for speaker verification. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6683–6687. IEEE (2021)
14. Nagrani, A., Chung, J.S., Zisserman, A.: Voxceleb: A large-scale speaker identification dataset. CoRR **abs/1706.08612** (2017), https://arxiv.org/abs/1706.08612
15. Okabe, K., Koshinaka, T., Shinoda, K.: Attentive statistics pooling for deep speaker embedding. CoRR **abs/1803.10963** (2018), https://arxiv.org/abs/1803.10963
16. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. Digital signal processing **10**(1-3), 19–41 (2000)
17. Togneri, R., Pullella, D.: An overview of speaker identification: Accuracy and robustness issues. IEEE circuits and systems magazine **11**(2), 23–61 (2011)
18. Wang, F., Cheng, J., Liu, W., Liu, H.: Additive margin softmax for face verification. IEEE Signal Process. Lett. **25**(7), 926–930 (2018)

19. Wang, H., Li, H., Li, B.: Vot: Revolutionizing speaker verification with memory and attention mechanisms. arXiv preprint arXiv:2312.16826 (2023)
20. Yu, Y.Q., Li, W.J.: Densely connected time delay neural network for speaker verification. In: Interspeech. pp. 921–925 (2020)
21. Zhang, Z.: Improved adam optimizer for deep neural networks. In: 2018 IEEE/ACM 26th international symposium on quality of service (IWQoS). pp. 1–2 (2018)