*INTERSPEECH 2020*
October 25–29, 2020, Shanghai, China

# ARET: Aggregated Residual Extended Time-delay Neural Networks for Speaker Verification

*Ruiteng Zhang*[1], *Jianguo Wei*[1,2], *Wenhuan Lu*[1], *Longbiao Wang*[1], *Meng Liu*[1], *Lin Zhang*[1], *Jiayu Jin*[1], *Junhai Xu*[1]

[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Computer College, Qinghai Nationalities University, Xining, China
{rtz, jianguo, Wenhuan, longbiao_wang, liumeng2017, zlinsmile, jinjiayu, jhxu}@tju.edu.cn

## Abstract

The time-delay neural network (TDNN) is widely used in speaker verification to extract long-term temporal features of speakers. Although common TDNN approaches well capture time-sequential information, they lack the delicate transformations needed for deep representation. To solve this problem, we propose two TDNN architectures. RET integrates short-cut connections into conventional time-delay blocks, and ARET adopts a split-transform-merge strategy to extract more discriminative representation. Experiments on VoxCeleb datasets without augmentation indicate that ARET realizes satisfactory performance on the VoxCeleb1 test set, VoxCeleb1-E, and VoxCeleb1-H, with 1.389%, 1.520%, and 2.614% equal error rate (EER), respectively. Compared to state-of-the-art results on these test sets, RET achieves a 23% $\sim$ 43% relative reduction in EER, and ARET reaches 32% $\sim$ 45%.

**Index Terms**: residual transformations, aggregated transformations, time-delay neural networks, speaker verification

## 1. Introduction

Speaker verification is an open-set task that determines whether two speeches are spoken by the same speaker. In conventional speaker verification, a speech utterance or speaker is represented by a fixed low-dimensional feature vector, such as an i-vector extracted by a Gaussian mixture model (GMM) [1]. LDA/PLDA [2] and cosine similarity can be used as a back-end function to calculate similarity scores of embeddings for different speakers. Two speeches are determined to be spoken by the same speaker if their similarity exceeds a threshold.

Unfortunately, conventional speaker verification assumptions often do not hold in real-world applications [3, 4, 5, 6]. To address this issue, end-to-end speaker verification utilizes neural networks as a feature extractor to extract robust embeddings in complex scenarios. The d-vector [7] firstly introduces a deep neural network (DNN) to learn the frame-level speaker representation. To obtain more robust embeddings, [8] proposed the x-vector system, which uses a time-delay neural network (TDNN) to produce features by considering the long-term temporal dependencies of speeches. In the deep speaker system [9] based ResNet [10], the residual transformations help networks to learn deep characteristics for speakers.

End-to-end speaker verification models are augmented by large margin softmax losses [11, 12, 13, 14, 15, 4]. The idea is to enhance the discrimination of embeddings by enlarging the inter-class distance and reducing the intra-class variation.

The residual transformations allow for building a very deep ResNet to enhance the discrimination of recordings. ResNet processes speech features as images to extract high-dimensional representations. However, its architecture shows limited performance at speaker verification because the specialized convolution kernel (such as $3 \times 3$ or $5 \times 5$) does not process features by considering complete dimensions of frequency. Although TDNN extracts features by considering the temporal dynamics of the signal, it lacks the capacity for deep representation. Ideal speaker-verification networks should well capture time-sequential information and extract the deep representation of speakers.

To improve the deep representation of TDNN, the extended time-delay neural network (E-TDNN) [16] extends TDNN layers by inserting dense layers between two TDNN layers. SpeakerResNet and similar structures [17, 18, 19] attempt to introduce a short-cut connection to TDNN, but they seem to lack robustness in complex scenarios. This is mainly because SpeakerResNet adopts a deep and narrow network architecture, which is not good at modeling personalized features of speakers.

In this paper, we propose two strategies to enhance deep representation based on modeling long-term temporal dependencies to improve the performance of TDNN. The first approach, the residual extended time-delay neural network (RET), integrates residual transformations into conventional time delay blocks. The aggregated residual extended time-delay neural network (ARET) builds a more delicate block, which combines the split-transform-merge strategy with residual transformations in the TDNN block. The strategy helps TDNNs to learn the more delicate relationships between frames of speeches to augment the discrimination of speakers. In our experiments, models are trained on the VoxCeleb2 dev set, and are evaluated on the VoxCeleb1 test set and the extended and hard test sets (VoxCeleb1-E and VoxCeleb1-H, respectively). The proposed ARET demonstrates significant performance advantages over state-of-the-art systems on these three test sets.

The remainder of the paper is organized as follows. Related works are briefly introduced in section 2. Section 3 proposes two modules to improve the deep representations of TDNN. The experimental setup is outlined in section 4. Experimental results are presented and analyzed in section 5. Finally, the last section draws a conclusion.

## 2. Related Work

### 2.1. Extended Time-delay Neural Networks

In speaker verification, TDNN extracts robust embeddings by capturing the long-term temporal relations of utterances. To improve the deep representation of features, E-TDNN interleaves dense layers between the TDNN layers. Similar to TDNN, it uses statistics pooling to aggregate frame-level features to segment-level features, which are passed to fully connected (fc)



http://dx.doi.org/10.21437/Interspeech.2020-1626

Table 1: *Architectures for TDNN, E-TDNN, RET, and ARET. T is the number of training segment frames, and N is the number of speakers. In this table, N is 1000 (to compare with ResNet intuitively), and the size of the input feature of networks is 161 × 300.*

| Layer Type | TDNN | | E-TDNN | | RET-17 | | ARET-21 | | ARET-25 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Context | Size | Context | Size | Context | Size | Context | Size | Context | Size |
| Frame1 | t-2:t+2 | 512 | t-2:t+2 | 512 | t-2:t+2 | 512 | t-2:t+2 | 512 | $\left\{\begin{bmatrix} t-2:t+2 \\ t \\ t-1:t+1, C=32 \\ t \end{bmatrix}\right\}\times 2$ | 512 |
| Frame2 | t-2, t, t+2 | 512 | t | 512 | $\begin{bmatrix} t-1:t+1 \\ t-1:t+1 \end{bmatrix}$ | 512 | $\begin{bmatrix} t \\ t-1:t+1, C=32 \\ t \end{bmatrix}$ | 512×2 | | 512×2 |
| Frame3 | t-3, t, t+3 | 512 | t-1:t+1 | 512 | t-1:t+1 | 512 | | 512 | | 512 |
| Frame4 | t | 512 | t | 512 | $\begin{bmatrix} t-1:t+1 \\ t-1:t+1 \end{bmatrix}$ | 512 | t-1:t+1 | 512 | t-1:t+1 | 512 |
| Frame5 | t | 512×3 | t-1:t+1 | 512 | | 512 | $\begin{bmatrix} t \\ t-1:t+1, C=32 \\ t \end{bmatrix}$ | 512×2 | $\begin{bmatrix} t \\ t-1:t+1, C=32 \\ t \end{bmatrix}$ | 512×2 |
| Frame6 | - | - | t | 512 | $\begin{bmatrix} t-1:t+1 \\ t-1:t+1 \end{bmatrix}$ | 512 | | 512 | | 512 |
| Frame7 | - | - | t-2:t+2 | 512 | t-1:t+1 | 512 | t-1:t+1 | 512 | t-1:t+1 | 512 |
| Frame8 | - | - | t | 512 | $\begin{bmatrix} t-1:t+1 \\ t-1:t+1 \end{bmatrix}$ | 512 | $\begin{bmatrix} t \\ t-1:t+1, C=32 \\ t \end{bmatrix}$ | 512×2 | $\begin{bmatrix} t \\ t-1:t+1, C=32 \\ t \end{bmatrix}$ | 512×2 |
| Frame9 | - | - | t | 512 | | 512 | | 512 | | 512 |
| Frame10 | - | - | t | 512×3 | t-2:t+2 | 512 | t-1:t+1 | 512 | t-1:t+1 | 512 |
| Frame11 | - | - | - | - | $\begin{bmatrix} t-1:t+1 \\ t-1:t+1 \end{bmatrix}$ | 512 | $\begin{bmatrix} t \\ t-1:t+1, C=32 \\ t \end{bmatrix}$ | 512×2 | $\begin{bmatrix} t \\ t-1:t+1, C=32 \\ t \end{bmatrix}$ | 512×2 |
| Frame12 | - | - | - | - | | 512 | | 512 | | 512 |
| Frame13 | - | - | - | - | t | 512 | t-2:t+2 | 512 | t-2:t+2 | 512 |
| Frame14 | - | - | - | - | t | 512×3 | t | 512 | t | 512 |
| Frame15 | - | - | - | - | - | - | $\begin{bmatrix} t \\ t-1:t+1, C=32 \\ t \end{bmatrix}$ | 512×2 | $\begin{bmatrix} t \\ t-1:t+1, C=32 \\ t \end{bmatrix}$ | 512×2 |
| Frame16 | - | - | - | - | - | - | | 512 | | 512 |
| Frame17 | - | - | - | - | - | - | t | 512 | t | 512 |
| Frame18 | - | - | - | - | - | - | t | 512×3 | t | 512×3 |
| Statistics pooling | [0,T] | 512×6 | [0,T] | 512×6 | [0,T] | 512×6 | [0,T] | 512×6 | [0,T] | 512×6 |
| Segment1 | [0,T] | 512 | [0,T] | 512 | [0,T] | 512 | [0,T] | 512 | [0,T] | 512 |
| Segment2 | [0,T] | 512 | [0,T] | 512 | [0,T] | 512 | [0,T] | 512 | [0,T] | 512 |
| AAM-Softmax | [0,T] | N | [0,T] | N | [0,T] | N | [0,T] | N | [0,T] | N |
| # params | $5.4 \times 10^6$ | | $7.8 \times 10^6$ | | $13.0 \times 10^6$ | | $10.1 \times 10^6$ | | $12.2 \times 10^6$ | |
| PLOPs | $0.9 \times 10^9$ | | $1.6 \times 10^9$ | | $3.1 \times 10^9$ | | $2.3 \times 10^9$ | | $2.9 \times 10^9$ | |

layers and the softmax output layer. Recently, E-TDNN [16] has performed compellingly at speaker verification. The architectures of TDNN and E-TDNN are shown in Table 1.

### 2.2. Residual Transformations in ResNet

ResNet [10] adopts residual transformations in a convolutional neural network (CNN) to solve the phenomenon of vanishing or exploding gradients and degradation problems; it achieves superior performance at many tasks. ResNet is also widely applied in speaker verification as a feature extractor to extract deep embeddings. The residual learning can be defined as:

$$y = F(x, \{W\}) + x, \tag{1}$$

where $x$ is the input feature, and $y$ is the output feature. $F(x, \{W\})$ represents the residual mapping to be trained.

### 2.3. Additive Angular Margin Loss

To enhance the representation of embeddings, the additive angular margin softmax loss (AAM-Softmax) [15] is used to increase the inter-speaker difference and reduce the intra-speaker variation by introducing an additive angular margin in the softmax loss. The equation of AAM-Softmax loss is:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(cos(\theta_{y_i}+m))}}{e^{s(cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{c}e^{s(cos\theta_j)}}, \tag{2}$$

where $N$ is the number of training samples, $c$ is the number of speakers in the training set, $m$ is the additive angular margin, and $s$ is a scaling factor.

## 3. Residual Transformations in TDNN

Conventional TDNN lacks the capacity to model deep representations of speeches, and ResNet does not capture long-term information. The idea TDNN should consider time-sequential information and have an efficient deep representation. To construct an efficient TDNN, we propose two robust strategies to introduce residual transformations in TDNN: the residual extended time-delay neural network (RET) and aggregated residual extended time-delay neural network (ARET). The structures of standard RET and ARET are shown in Table 1 (the figure following the name of a model denotes its number of layers).

### 3.1. Residual Extended Time-delay Neural Networks

We preset the residual TDNN (RT) block (depicted in Figure 2 (left)), which ensures TDNN can be served by residual learning with identity mapping. Each RT block has TDNN layers, batch normalization, and LeaklyReLU activation functions.

RET uses RT blocks with temporal context $\{t-1, t, t+1\}$ to replace the extended layers in E-TDNN. RET efficiently combines the ability to capture temporal features with the capacity to extract a deep representation of features. As shown in the third part of Table 1, standard RET has 14 frame-level layers, including four RT blocks. After obtaining frame-level features, it uses statistics pooling to aggregate frame-level features to segment-level. Then segment-level features are fed into two fc networks, and the last fc layer can be connected with the AAM-Softmax loss function.

### 3.2. Aggregated Residual Extended Time-delay Neural Networks

To build more efficient architectures with more delicate transformations, we present aggregated residual TDNN (ART) blocks based on aggregated transformations [20] to enhance the delicate deep representation strength of TDNN. Aggregated transformations in TDNN can be presented as:

$$T(x) = \sum_{i=1}^{C} T_{sub}(x, i), \tag{3}$$

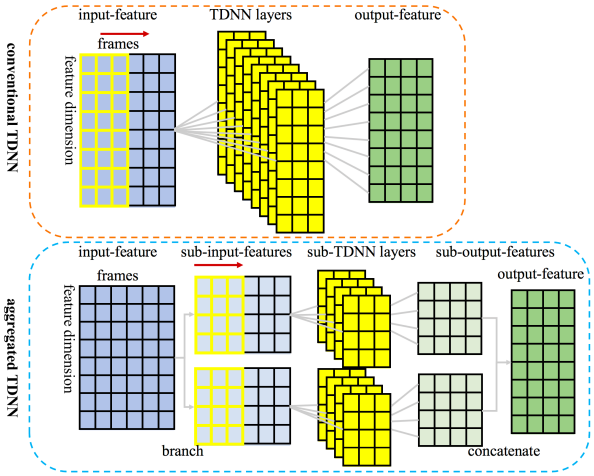


Figure 1: *TDNN with and without aggregated transformations. Note that the TDNN layer with aggregated transformations only has 1/C of the parameters of the conventional TDNN layer.*

where $T_{sub}(x, i)$ is the $i$-th sub-TDNN neuron, and $C$ is the size of the set of transformations to be aggregated. TDNN with and without aggregated transformations are displayed in Figure 1. The input feature is split into $C$ sub-features, which are passed on to corresponding sub-TDNN layers with the same topology. Each sub-outputs are concatenated as the output of the transformations . The split-transform-merge behavior of TDNN outperforms the representational power of large and dense layers, with considerably less computational complexity.

The aggregated transformations in TDNN can be combined with the residual transformations and defined as:

$$y = x + \sum_{i=1}^{C} T_{sub}(x, i), \tag{4}$$

where $x$ is the input feature, $y$ is the output, and $T_{sub}(x, i)$ is the $i$-th sub-TDNN layer.

Figure 2 (right) shows an ART block. The feature from the preceding layer is fed into $C$ paths sharing the same topology. Each path includes three TDNN layers, which have $\{t\}$, $\{t-1 : t+1\}$, $\{t\}$ context, successively. The output of the ART block is connected to the sub-outputs of each path, which can be served with residual transformations. First, the split-transform-merge strategy in TDNN helps isolate a few factors and keeps the temporal sequential information of each path to provide more delicate representations of TDNN. In an ART block, therefore, different sub-ways that have relatively independent relationships can focus on the various frequency information to help the network to distinguish speakers well. Then the strategy improves the efficiency of weights of layers to reduce the parameters. Compared to RET, ARET significantly reduces the parameters and floating-point operations (FLOPs), and has more delicate representations for speakers.

The architecture of the standard ARET (ARET-21) is shown in Table 1. Similar to [20] constructed aggregated transformations by grouped convolutional layers [21], we implement ARET by grouped TDNN layers. ARET has 18 frame layers, including four ART blocks. RET and ARET use two efficient methods to make the TDNN obtain deep representational capability similar to ResNet, but there are many fewer parameters

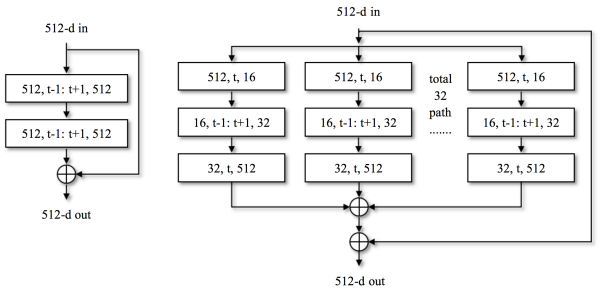


Figure 2: ***(Left):*** *A residual TDNN block with identity mapping of RET.* ***(Right):*** *An aggregated residual TDNN block with $C = 32$ of ARET, with a bottleneck architecture. A layer is denoted as (# in channels, context of layers, # out channels).*

(ResNet-34 with a 1000-dimensional fc layer has around 21.8M parameters, while the proposed models have fewer than 13M).

# 4. Experimental setup

## 4.1. Dataset

To investigate the performance of the proposed models, we ran experiments on the VoxCeleb1 [5] and VoxCeleb2 [6] datasets without any data augmentation. The VoxCeleb2 dev set was used to train models. The VoxCeleb1 test part, VoxCeleb1-E [6], and VoxCeleb1-H [6] were used as evaluation sets. VoxCeleb1-E and VoxCeleb1-H are composed by VoxCeleb1 dev and test sets. Besides, VoxCeleb1-H is a hard condition that testing on the pairs with the same nationality and gender. The configuration of these databases is summarized in Table 2. Experiments utilized the equal error rate (EER), minimum detection cost function from NIST SRE08 (minDCF08) [22] and SRE10 (minDCF10) [23] as the performance metrics.

Table 2: *Configuration of databases.*

| Training Set | # Speakers | # Utterances | Testing Set | # Speakers | # Pairs |
|---|---|---|---|---|---|
| VoxCeleb1 dev | 1,211 | 148,642 | VoxCeleb1 test | 40 | 37,719 |
| VoxCeleb2 dev | 5,994 | 1,092,009 | VoxCeleb1-E | 1,251 | 581,480 |
| | | | VoxCeleb1-H | 1,190 | 552,536 |

## 4.2. Experimental details

**Networks.** The experiments used TDNN, E-TDNN, RET, and ARET to extract deep embeddings from spectrograms. To investigate the performance of RET and ARET more comprehensively, we added a TDNN layer and ART block to ARET-21 to produce ARET-25, which has similar parameters and FLOPs to RET-17. ARET-25 is shown in Table 1.

**Feature.** 161-dimensional spectrograms that were generated in sliding window fashion from hamming window of width 25 ms and step 10 ms were adopted as input features. No voice activity detection (VAD) or augmentation was applied.

**Hyperparameter.** The margin $m$ and scale factor $s$ for AAM-Softmax loss were set to 0.25 and 30, respectively. $C$ in ARET was set to 32.

**Training.** In the training stage, mini-batch size of 64 was used to train models. Each training speech sample was randomly sampled for 300 frames from each audio. Stochastic gradient descent (SGD) with momentum 0.9, weight decay 0.00001, and initial learning rate 0.1 was utilized.

Table 3: *Comparison of performance of two proposed models with other speaker-verification networks on the original VoxCeleb1 test set and extended and hard test sets (VoxCeleb1-E and VoxCeleb1-H).*

| Description | Front-end model | Training dataset | Loss function | EER(%) | minDCF08 | minDCF10 |
|---|---|---|---|---|---|---|
| **VoxCeleb1 test set** | | | | | | |
| Chung *et al.* [6] | ResNet-50 | VoxCeleb2 | Softmax + Contrastive | 4.190 | - | - |
| Xie *et al.* [24] | Thin ResNet-34 | VoxCeleb2 | Softmax | 3.220 | - | - |
| Xiang *et al.* [25] | TDNN | VoxCeleb2 | AAM-Softmax | 2.694 | - | - |
| Liu *et al.* [4] | TDNN | VoxCeleb1+VoxCeleb2 | AAM-Softmax | 2.030 | 0.0120 | 0.4010 |
| Our implementation | TDNN | VoxCeleb2 | AAM-Softmax | 2.341 | 0.0120 | 0.3270 |
| Proposed | RET-17 | VoxCeleb2 | AAM-Softmax | 1.559 | 0.0083 | **0.1981** |
| Proposed | ARET-25 | VoxCeleb2 | AAM-Softmax | **1.389** | **0.0072** | 0.1987 |
| **VoxCeleb1-E** | | | | | | |
| Chung *et al.* [6] | ResNet-50 | VoxCeleb2 | Softmax + Contrastive | 4.420 | - | - |
| Nagrani *et al.* [26] | Thin-ResNet-34 | VoxCeleb2 | Softmax | 2.950 | - | - |
| Xiang *et al.* [25] | TDNN | VoxCeleb2 | AAM-Softmax | 2.762 | - | - |
| Our implementation | TDNN | VoxCeleb2 | AAM-Softmax | 2.290 | 0.0115 | 0.3933 |
| Proposed | RET-17 | VoxCeleb2 | AAM-Softmax | 1.599 | 0.0077 | 0.3016 |
| Proposed | ARET-25 | VoxCeleb2 | AAM-Softmax | **1.520** | **0.0071** | **0.2700** |
| **VoxCeleb1-H** | | | | | | |
| Chung *et al.* [6] | ResNet-50 | VoxCeleb2 | Softmax + Contrastive | 7.330 | - | - |
| Nagrani *et al.* [26] | Thin-ResNet-34 | VoxCeleb2 | Softmax | 4.930 | - | - |
| Xiang *et al.* [25] | TDNN | VoxCeleb2 | AAM-Softmax | 4.732 | - | - |
| Our implementation | TDNN | VoxCeleb2 | AAM-Softmax | 3.849 | 0.0182 | 0.5077 |
| Proposed | RET-17 | VoxCeleb2 | AAM-Softmax | 2.700 | 0.0127 | 0.4004 |
| Proposed | ARET-25 | VoxCeleb2 | AAM-Softmax | **2.614** | **0.0122** | **0.3878** |

**Testing.** Full-length utterances were adopted in the testing stage to extract embeddings. Cosine similarity was applied as the back-end scoring method.

## 5. Results

Experimental results on the VoxCeleb1 test part, VoxCeleb1-E, and VoxCeleb1-H are shown in Table 3. Past state-of-the-art systems were chosen as baselines to gauge the performance of the two presented modules.

The first proposed RET-17 achieved 23% to 43% relative EER reduction on three test sets, with results of 1.559%, 1.599%, and 2.700% EER on the VoxCeleb1 test part, VoxCeleb1-E, and VoxCeleb1-H, respectively.

The presented ARET-25 achieved state-of-the-art performance, with 1.389% EER, 0.0071 minDCF08, and 0.1987 minDCF10 on the VoxCeleb1 test part. Compared with the best baseline in this part, [4] adopted VoxCeleb1 and VoxCeleb2 to train x-vector while our ARET-25 only trained on VoxCeleb2 dev part, ARET-25 had 32%, 40%, and 50% relative decrease. On VoxCeleb1-E and VoxCeleb1-H, decreased EER to 1.520% and 2.614%, respectively, which are both 45% relatively lower than the best results in [25].

Figure 3 shows the detection error tradeoff (DET) curves of TDNN, E-TDNN, and the proposed models (RET-17, ARET-21, ARET-25). Compared to RET-17, ARET-25 reached state-of-the-art performance, which relatively decreased 3% to 11% evaluation metrics but with less parameters and FLOPs. These results show that our ART blocks provide more delicate representations of ARET, to help obtain stronger discriminations of speakers.

## 6. Conclusion

This paper presents two strategies to replace ResNet and TDNN as the feature extractor in end-to-end-based speaker verification.

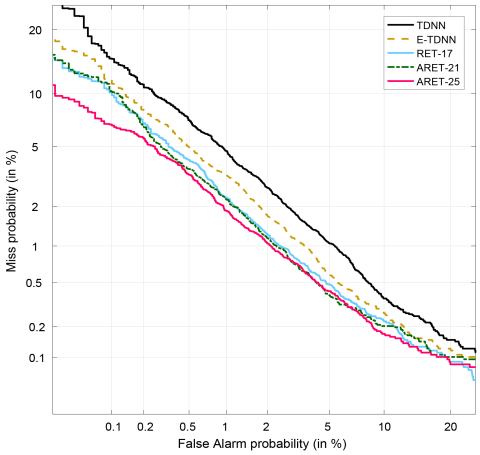


Figure 3: *Comparison of TDNN, E-TDNN, RET-17, ARET-21, and ARET-25 using DET curve for original VoxCeleb1 test part.*

RET and ARET respectively introduce residual and aggregated residual transformations in TDNN. The best proposed ARET achieves state-of-the-art performance on the VoxCeleb1 test set, VoxCeleb1-E, and VoxCeleb1-H, obtaining 32% to 45% relative EER reduction compared to strong baselines. These competitive results demonstrate that RET and ARET have the potential to replace TDNN or ResNet in speaker verification.

## 7. Acknowledgements

This work was supported by the National Key R&D Program of China (No. 2018YFC0806802), National Natural Science Foundation of China (No. 61876131, U1936102).

# 8. References


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

[2] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. International Conference on Computer Vision*, 2007, pp. 1–8.

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

[4] Y. Liu, L. He, and J. Liu, "Large Margin Softmax Loss for Speaker Verification," in *Proc. Interspeech*, 2019, pp. 2873–2877.

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

[6] A. N. Joon Son Chung and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.

[7] 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. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4052–4056.

[8] 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.

[9] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. ICML*, 2016, pp. 507–516.

[12] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proc. IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.

[13] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[14] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.

[15] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[16] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5796–5800.

[17] S. Novoselov, A. Shulipa, I. Kremnev, A. Kozlov, and V. Shchemelinin, "On deep speaker embeddings for text-independent speaker recognition," *arXiv preprint arXiv:1804.10080*, 2018.

[18] G. Bhattacharya, J. Alam, and P. Kenny, "Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6041–6045.

[19] S. Novoselov, A. Gusev, A. Ivanov, T. Pekhovsky, A. Shulipa, G. Lavrentyeva, V. Volokhov, and A. Kozlov, "STC Speaker Recognition Systems for the VOiCES from a Distance Challenge," in *Proc. Interspeech*, 2019, pp. 2443–2447.

[20] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[21] G. E. Hinton, A. Krizhevsky, and I. Sutskever, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1106–1114, 2012.

[22] A. F. Martin and C. S. Greenberg, "NIST 2008 speaker recognition evaluation: Performance across telephone and room microphone channels," in *Proc. Interspeech*, 2009, pp. 2579–2582.

[23] ——, "The NIST 2010 speaker recognition evaluation," in *Proc. Interspeech*, 2010, pp. 2726–2729.

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

[25] X. Xiang, S. Wang, H. Huang, Y. Qian, and K. Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," *arXiv preprint arXiv:1906.07317*, 2019.

[26] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.