# IBG_AI Speaker Recognition System for Far-Field Speaker Verification Challenge 2020

*Feifei Zhou,Chuan Ke, Mingqing Zhu, Yi Peng, Fei Wu*

International Business Group, Tencent Inc., Shenzhen, China

{feifeizhou,jayke,migosszhu,yipeng,flyingwu}@tencent.com

## Abstract

In this report, we describe the submissions of International Business Group AI (IBG_AI) team to Far-Field Speaker Verification Challenge 2020. Our submitted system for task 2 is a fusion of four models based on ResNet with Squeeze-Excitation. The first two networks have ResNet152 topologies with or without voice activity detections (VAD), while the last two models use ResNet203. As for task 1, only the two ResNet203 model is used to obtain the fusion model. 90-dimensional FBank features are extracted as the input of models, and all the models are trained by using softmax loss and additive angular margin (AAM) loss during train and fine-tune stages respectively. Besides, we adjusted the proportion of training data and fine-tuned some extra steps. Finally, adaptive symmetric score normalization is applied to normalize scores. As a result, our best systems for task1 and task2 achieved 0.372 minDCF and 0.317 minDCF on the challenge evaluation set respectively.

**Index Terms**: speaker verification, speaker recognition, far-field, deep neural network

## 1. Introduction

Recently, speaker recognition has received an increasing amount of interests in smart applications. Although it has made a great progress with the development of deep learning and the availability of large-scale datasets, yet there are still significant challenges. The method should be able to generalize well in order to be robust to noisy, far-field and new possible deployment conditions.

Deep speaker embedding based systems[1, 2] (like x-vectors) have been shown to significantly outperform conventional i-vector[3] based systems in terms of speaker recognition performance. And BUT system[4] showed that the deeper network such as ResNet160 is more effective than Time Delay Neural Network (TDNN)[2]. In addition, some methods from face recognition field were used for speaker recognition[5]. A comparative study of different loss functions for DNN based speaker embeddings was presented in [6].

This report describes the IBG_AI speaker recognition systems submitted to task1 and task2 in FFSVC 2020[7]. In general, the speaker recognition system can be divided into three parts, the front-end, the speaker embedding and the back-end. The rest of this document is organized as follows: in Section 2, we describe the details of each part of our system, in Section 3, several experiments are conducted to get the result, and the final conclusion are presented in Section 4.

## 2. System components description

In this section, we introduce all the components used in our systems.

### 2.1. Front-End

#### *2.1.1. Training data, Augmentations*

FFSVC 2020 challenge dataset includes Chinese Mandarin audios received at different distances, more information about the dataset can be found in [7]. And the goal of task1 and task 2 is to determine whether an audio pair which consists of a near-field (at 25 cm) and a far-filed audio is from the same speaker, the difference of the two tasks is that task 1 focus on text-dependent audios while task2 uses text-independent recordings.

As any publicly open and freely accessible dataset shared on openslr.org[1] before the challenge is allowed to use, we selected 7 datasets for training. The train set consists of SLR33 (Aishell-1)[8], SLR38[9], SLR47[10], SLR62[11], SLR68[12], SLR49[13] and the train data FFSVC 2020 provided. After remove the speaker that doesn't have enough audios, the train set contains 10674 speakers in total.

Besides, SLR17 (MUSAN)[14] and SLR28 (Room Impulse Response and Noise Database, RIRs)[15] are also used to argument the train data. The augmentation process was based on the Kaldi recipe[2]and it resulted in additional 4 times utterances belonging to the following categories: reverberated using RIRs, augmented with Musan noise, music, and babel.

#### *2.1.2. Features and VAD*

All of our models make use of FBank features, the 90-dimensional FBank features are extracted in the way similar to BUT system[4]. It is extracted from audios which is downsampled to 16kHz, its frequency is limited to 20-7600Hz, and the frame length is 25ms with 10ms shift. As audios in different dataset have various volume levels, and there is also a large volume gap between audios received at different distances, the FBank feature is normalized by subtracting the mean and divided by the standard deviation with a sliding window of 3 seconds.

As for VAD, two of our four models make use of it, while the other two not. A basic energy-based VAD in Kaldi is applied, and only the valid frames are selected.

### 2.2. Speaker embedding

#### *2.2.1. Model structure*

The backbone network of our system is the well-known ResNet topology[16]. Inspired by BUT system, we halved the number of channels of each ResNet block, thus can get a deeper network with the similar number of model parameters. The details of the ResNet152 topology is shown in Table 1, while in ResNet203, the number of layers in each ResNet block is changed to [3, 12, 49, 3] .

[1]https://openslr.org/index.html

[2]https://github.com/kaldi-asr/kaldi

In addition, attentive statistics pooling layer[17] is utilized to aggregate frame-level representation on utterance level, benefit from the attentional mechanism, the model is able to assign bigger weights to more important frames when calculate statistics, which makes the model more robust to the external environment interference and silence frames.

In our models, the shape of the last ResNet block output is $(d_1, T_1, c)$, $c$ is the number of channels, $d_1$ and $T_1$ represent the number of feature dimensions and frames after they pass through ResNet blocks respectively. In order to reduce the attention network parameters, the output is averaged along FBank feature dimension, which leads to a matrix $H$ of shape $(T_1, c)$, and then additive attention with two fully-connected layers is utilized to get the weights of every frames $e_t$:

$$e_t = \mathrm{softmax}(f(f(h_t))) \tag{1}$$

where $f(.)$ is fully connected layer with nonlinear activation function, $h_t$ is each row of $H$. Finally, the weighted mean and standard deviation of each dimension is concatenate together as the input of the rest network.

Besides, as the model benefits from a wider temporal context, it could be beneficial to rescale the frame-level features given global properties of the recording. For this purpose, 2-dimensional Squeeze-Excitation (SE) blocks is introduced to the get the weights of each channel[18] in the last two ResNet block as shown in Table 1. The first component of an SE-block is the freeze operation which calculates the mean of each channel:

$$\boldsymbol{z} = \frac{1}{T_0 d_0} \sum_{t=1}^{T_0} \sum_{i=1}^{d_0} \boldsymbol{u}_{t,i} \tag{2}$$

where $\boldsymbol{u}_{t,i}$ is the vector consists of every channel data at time $t$ and feature $I$, $d_0$ and $T_0$ represent the number of feature dimensions and frames after passing through the convolution network in ResNet block respectively. Then two-layer fully connected layer is used to get the weights of each channel:

$$\boldsymbol{s} = \sigma(f(f(\boldsymbol{z}))) \tag{3}$$

where $\sigma$ denotes the sigmoid function, and the resulting vector $\boldsymbol{s}$ contains weights $s_c$ between 0 and 1, which are applied to the original input by channel-wise multiplication.

As shown in Table 1, speaker embeddings (512 dimensional) are extracted from the last batch normalization layer (Dense1-Relu-BN in Table1). There is no bias used in the last fully connected layer, so the weights of it can be regarded as the embedding centers of speakers in train set. These speakers embedding center is used as cohort in score normalization.

#### *2.2.2. Loss Function*

We trained the raw model with Softmax cross entropy loss, and then fine-tune it with AAM loss. AAM loss was proposed for face recognition, it introduces a large margin $m$ to improve the intra-class compactness and inter-class discrepancy:

$$L_{AAM} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{s(\cos(\theta_{y_i} + m))}}{e^{s(\cos(\theta_{y_i} + m))} + \sum_{j=1, j \neq y_i}^{C} e^{s \cos \theta_j}} \tag{4}$$

where $\theta_i$ is the angle between the $i$-th class center and the embedding, $s$ is the scale factor and $C$ is the number of speakers. In all of our experiments, $m$ is set to 0.25 and $s$ is set to 30.

Table 1: *The proposed ResNet152 architecture. $C$ in the last row is the number of speakers. $S$ in Structure column indicates the stride of convolution. The first dimension of the input shows number of filter-banks and the second dimension indicates the number of frames*

| **Layer name** | **Structure** | **Output** |
|---|---|---|
| Input | - | $(90, L, 1)$ |
| Conv2D-1 | $3 \times 3$,$S$=1 | $(90, L, 32)$ |
| ResNetBlock1 | $\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 32 \end{bmatrix} \times 3$, $S$=1 | $(90, \frac{L}{2}, 128)$ |
| ResNetBlock2 | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 32 \end{bmatrix} \times 8$, $S$=2 | $(45, \frac{L}{4}, 256)$ |
| ResNetBlock3 | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 32 \\ \mathrm{SE-Block} \end{bmatrix} \times 36$, $S$=2 | $(23, \frac{L}{6}, 512)$ |
| ResNetBlock4 | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 32 \\ \mathrm{SE-Block} \end{bmatrix} \times 3$, $S$=2 | $(12, \frac{L}{8}, 1024)$ |
| Att-StatPooling | - | $(24, 1024)$ |
| Flatten | - | 24576 |
| Dense1-Relu-BN | - | 512 |
| Dense2 | - | $C$ |

### 2.3. Back-End

The cosine distance was used to discriminate speakers. In addition, adaptive symmetric score normalization (AS-norm)[19] which computes an average of normalized scores from Z-norm[20] and T-norm[21] was adopted. The cohort was created by using the weights of the last fully connected layer as speaker center. It consisted of 10674 speaker centers. In AS-norm, only part of the cohorts are selected to compute mean and variance for normalization. Usually $X$ top scoring or most similar cohorts are selected; $X$ was set to ten percent of total numbers of speakers for all experiments.

### 2.4. Fusion

For score level, fusion was performed by computing the average of the scores of the individual systems.

## 3. Results and Discussion

At every training step, we random sample 32 speakers, and for each speaker, 4 segments which are randomly cropped between 250 frames and 300 frames are selected, if the length of a audio is smaller than 250 frames, we repeat it util reach the minimum length, i.e. 250 frames. Softmax with cross entropy is used to train the raw system, and then AAM loss is utilized to fine-tune the model. In all training procedure, we select stochastic gradient descent (SGD) as the optimizer and the weight decay is set to 0.0002 in Pytorch. The initial learning rate to train the raw model is set to 1.0 while that to fine-tune the model is set

Table 2: *Results of our systems for FFSVC 2020 task 2 challenge with/without score normalization. The 40dims prefix indicates 40-dimensional FBank features, SE indicates Squeeze-Excitation Block*

| ID | VAD | Embd NN | Extra Fine-Tune | Development Set minDCF | Development Set EER | Evaluation Set minDCF | Evaluation Set EER |
|---|---|---|---|---|---|---|---|
| 1 | No | 40dims-ResNet152 | No | 0.4849/0.4778 | 3.31% /3.27% | - | - |
| 2 | No | ResNet152 | No | 0.4116/0.3869 | 3.16%/2.95% | - | - |
| 3 | No | ResNet152-SE | No | 0.3736/0.3462 | 2.86%/2.68% | - | - |
| 4 | No | ResNet152-SE | Yes | **0.3376/0.3188** | 2.27%/2.24% | - | - |
| 5 | No | ResNet203-SE | Yes | 0.3544/0.3263 | 2.3%/2.24% | - | - |
| 6 | Yes | ResNet152-SE | Yes | 0.3505/0.3377 | 2.23%/2.18% | - | - |
| 7 | Yes | ResNet203-SE | Yes | 0.3446/0.3218 | **2.03**%/**2.17**% | - | - |
| Fusion 4-7 | | | | 0.3222/0.2898 | 2.06%/2.03% | -/0.3165 | -/2.61% |

Table 3: *Results of our systems for FFSVC 2020 task 1 challenge with/without score normalization. SE indicates Squeeze-Excitation Block*

| ID | VAD | Embd NN | Extra Fine-Tune | Development Set minDCF | Development Set EER | Evaluation Set minDCF | Evaluation Set EER |
|---|---|---|---|---|---|---|---|
| 1 | No | ResNet152-SE | Yes | 0.4101/- | 2.8%/- | - | - |
| 2 | No | ResNet203-SE | Yes | **0.3327/0.3164** | **2.21**%/**2.11**% | - | - |
| 3 | Yes | ResNet152-SE | Yes | 0.436/- | 2.96%/- | - | - |
| 4 | Yes | ResNet203-SE | Yes | 0.36/0.3475 | 2.3%/2.29% | - | - |
| Fusion 2, 4 | | | | 0.3152/0.301 | 2.19%/2.13% | -/0.372 | -/3.17% |

to 0.1. During both training stages, learning rate halved every 10000 steps and the model is trained for 100000 steps in each stage.

In addition, in order to highlight FFSVC original training data, after fine-tune the model for 100000 steps, we adjusted the proportion of training data, that is, the sampled 32 speakers at each step consists of 16 speakers of FFSVC original data and 16 speakers of other datasets, and then fine-tine it for extra 10000 steps. It is referred as extra fine-tune in the following text. The difference of our systems for task 1 and task 2 is shown in this extra fine-tune stage. For task 2, the audios of fixed content "ni hao mi ya" are filtered out. While for task 1, only these fixed content part is retained as FFSVC original data, and the crop range is changed to [130, 150] frames. The following experiments show that this extra fine-tune can make an improvement.

The ResNet152 in our system has 39M parameters while ResNet203 has 44.6M parameters. Calculating the score of each trial by using ResNet152 will consumes 3.7GB memory of Tesla P40 GPU and takes about 250 milliseconds, and these costs for ResNet203 increase to 4.3GB memory and 285 milliseconds.

The results of the models for task 2 are displayed in Table 2, the fusion which makes use of Model 3, Model 4, Model 5 Model 6 is our best submission, and achieve 0.3165 minDCF and 2.61% EER on the challenge evaluation set. The scores are presented in the way without/with score normalization, for example, the scores 0.4849/0.4778 for model 1 means it is 0.4849 without score normalization and 0.4778 with score normalization. Score normalization is useful since it can get better results in all experiments. All models uses 90-dimensional Fbank features except model 1, it is obvious that 90-dimensional features is much better than 40-dimensional features by comparing model 1 and model 2. Besides, it also can be concluded that SE-block is helpful for the system by comparing model 2 and model 3. In addition, extra fine-tune can also make an improvement as model 4 is better than model 3. However, deepening the network and using VAD both have no obvious and stable effect for the system by analyzing model 4 to model 7.

The results of the models for task 1 are shown in Table 1, since ResNet203 works better than ResNet152 in task 1, the submitted fusion only combined model 2 and model 4, and got the result of 0.372 minDCF and 3.17% EER on the evaluation set.

## 4. Conclusions

In this work, we presented our speaker verification system for FFSVC 2020 task2 and task 1. Moreover, by the comparative experiments, it is concluded that higher dimensional input features, SE block, extra fine-tune and score normalization are beneficial for the system, while VAD not. Our final submission achieved 0.372 minDCF and 0.317 minDCF for task 1 and task 2 on the challenge evaluation set respectively.

## 5. References


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

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

[3] 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, 2010.

[4] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot,

"But system description to voxceleb speaker recognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[5] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, pp. 926–930, 2018.

[6] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," *arXiv preprint arXiv:1904.03479*, 2019.

[7] X. Qin, M. Li, H. Bu, R. K. Das, W. Rao, S. Narayanan, and H. Li, "The ffsvc 2020 evaluation plan," *arXiv preprint arXiv:2002.00387*, 2020.

[8] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell1: An open-source mandarin speech corpus and a speech recognition baseline," in *Oriental COCOSDA 2017*, 2017, p. Submitted.

[9] "Stcmds20170001_1, free st chinese mandarin corpus," https://www.surfing.ai.

[10] Primewords Information Technology Co., Ltd., "Primewords Chinese Corpus Set 1," 2018, https://www.primewords.cn.

[11] "aidatatang_200zh, a free Chinese Mandarin speech corpus by Beijing DataTang Technology Co., Ltd." www.datatang.com.

[12] "Magic Data Technology Co., Ltd." 2019, http://www.imagicdatatech.com/index.php/home/dataopensource/data_info/id/101.

[13] "Voxceleb data," https://openslr.org/49/.

[14] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[15] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

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

[17] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," *Proc. Interspeech 2018*, pp. 2252–2256, 2018.

[18] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

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

[20] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[21] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 42–54, 2000.