The Huawei System for 2020 Far-Field Speaker Verification Challenge

Jinwen Huang, Weixiang Hu, Yu Lu, Lei Miao, Renyu Wang, Zhuzi Chen, Huan Zhou

Huawei Technologies Co Ltd, China

huangjinwen2@huawei.com

Abstract

This report describes the systems submitted to the Far-Field Speaker Verification Challenge (FFSVC2020) [1][2] by our team, named as try123. For this speaker verification system, two types of end-to-end multi-channel model like ResNet and Res2Net are used as backbone model, and three types of layer like GhostVlad [11], global statistics pooling (GSP) and global statistic plus max pooling (GSPMP) are used as following encoding layer. The final fusion system integrated 6 models from different backbone models and encoding layers. Finally, the submitted evaluation trail results (30% of test set) on leaderboard are (minDCF 0.3152, EER 3.03%) for task1, (minDCF 0.3632, EER 3.03%) for task2 and (minDCF 0.2849, EER 3.06%) for task3.

Index Terms: speaker verification, far-field speech, ResNet, Res2Net, multi-channel

1. Introduction

Multi-channel training framework based on deep speaker embedding network like ResNet. Based on 2-dimensional (2D) or 3-dimensional (3D) convolution layer, the network is used to get the state of art performance for far-field speaker recognition under the reverberant and noisy environment with a multi-channel microphone array in [3]. We use multi-channel ResNet [4] and multi-channel Res2Net [5] for this challenge.

The following sections describes the details of our models and the fusion system.

2. Data usage

All training data comes from openslr.org and the FFSVC20 Challenge Dataset as list in following Table 1.

Table 1: *Datasets used for training models of the system*

Dataset	Identifier
Free ST Chinese Mandarin Corpus	SLR38
Aishell	SLR33
MAGICDATA Mandarin Chinese Read Speech Corpus	SLR68
Primewords Chinese Corpus Set1	SLR47
aidatatang_200zh	SLR62
CN-Celeb	SLR82
VoxCeleb Data	SLR49
LibriSpeech	SLR12
HI-MIA	SLR85
FFSVC20 Challenge Dataset	

There are two stages for our model training, pre-train and fine-tune. Training data include SLR38, SLR33, SLR68, SLR47, SLR62, SLR82, SLR49 and SLR12 are used in pre-train stage.

For task1 and task3, training data include HI-MIA (SLR85) and the text-dependent dataset from FFSVC 2020 are used in the fine-tune stage.

For task2, training data include HI-MIA (SLR85) and the text-independent part of FFSVC 2020 training dataset are used in the fine-tune stage. As HI-MIA (SLR85) is a text-dependent dataset, we use MultiReader method [6] to balance the training loss.

3. System description

3.1. Data augmentation

In pre-train stage, with pyroomacoustics toolkit [7] for simulating the room acoustic condition, 35% of the training data are randomly selected to generate far-field multi-channel data for model training.

Music, noise and speech part from MUSAN dataset [8] is used as additive noise with random SNR setting from 5db to 30db both in pre-training and fine-tune stage. In pre-train stage, noise is directly added in single-channel training data, and for multi-channel training data, pyroomacoustics toolkit is used for adding noise. In fine-tune stage, we only add noise to single-channel data.

The method of SpecAugment [9] is also applied in both pre-train and fine-tune stage.

Speed perturbation [10] used to get 3-times larger number of speaker IDs in fine-tune stage.

3.2. Acoustic Feature Extraction

All training are resampled to 16k Hz and pre-emphasized before feature extraction. The 64-dimensional Mel-log-filterbank energies is extracted with a frame length of 25ms and hop size of 10ms, and normalized through mean subtraction without voice activity detection.

3.3. Deep Speaker Embedding

Two different backbone were investigated: (1) ResNet34 and (2) Res2Net50. For each backbone, we use three different encoding layer: (1) GhostVlad [11], (2) global statistics pooling (GSP), (3) global statistic plus max pooling (GSPMP). Following encoding layer, a fully-connected layer is used to processes the utterance-level representation and finally get the speaker embedding after L2-normalization. Then we get six different models and integrate as the final fusion system.

To make full use of multi-channel data, we change the Conv and Batchnorm layers in the input stem and first stage of ResNet34 and Res2Net50 from 2d to 3d. For the single-channel data, we repeat the data four times to produce the multi-channel data. Furthermore, for matching the dimension between the 3D convolution feature maps (4D tensor) and 2D convolution feature maps (3D tensor), a 3D convolution layer with kernel size of $4 \times 1 \times 1$ is used between first stage and second stage as described in [3].

All the models are trained with angular softmax loss [12] in both pre-train and fine-tune stage.

3.4. Backend

In this work, cosine similarity is used for scoring without score normalization.

4. Experiment results

In pre-train stage, all models were trained with the training data describe in section 2, using Adam optimizer with constant learning rate as 0.001. Table 2 show the performance of six individual pre-train models on task2 dev dataset.

Table 2: *pre-train performance on task2 dev*

Model	EER (%)	minDCF
ResNet34 + GSP	5.4696	0.5453
ResNet34 + GSPMP	5.1945	**0.5450**
ResNet34 + GhostVlad	5.5599	0.6151
Res2Net50 + GSP	**5.1314**	0.5489
Res2Net50 + GSPMP	5.6631	0.5520
Res2Net50 + GhostVlad	5.8394	0.5548

In fine-tune stage, all models were trained with the training data describe in section 2, using Adam optimizer with the learning rate decreases from 0.0001 to 0 linearly. The final system is fused from the six individual models with score-level weighting, which is refined by different experiments. Table 3~5 show the performance of six individual models and final fusion system after fine-tune on task1 dev, task2 dev and task3 dev respectively.

In both pre-training and fine-tune stages, we used automatic search method for data augmentation and training hyper-parameters.

On task1, ResNet34 + GSP gets the best performance by minDCF, while Res2Net50 + GSP get the best performance by EER. On task2, ResNet34 + GhostVlad is the best model. On task 3, ResNet34 + GhostVlad and Res2Net + GSP obtain the best result by minDCF and EER respectively. From task1 and task3, we can see that the backbone of Res2Net50 performance better than ResNet34 by EER.

Table 3: *Fine-tune performance of each model and the final fusion system on task1 dev*

Model	EER (%)	minDCF
ResNet34 + GSP	2.3403	**0.2539**
ResNet34 + GSPMP	2.9183	0.3042
ResNet34 + GhostVlad	2.3673	0.2836
Res2Net50 + GSP	**2.0327**	0.287
Res2Net50 + GSPMP	2.1836	0.2567
Res2Net50 + GhostVlad	2.4489	0.3057
Fusion	**1.8535**	**0.2127**

Table 4: *Finetune performance of each single model and the fused system on task2 dev*

Model	EER (%)	minDCF
ResNet34 + GSP	3.1167	0.3734
ResNet34 + GSPMP	3.273	0.3841
ResNet34 + GhostVlad	**2.6685**	**0.3305**
Res2Net50 + GSP	3.268	0.3964
Res2Net50 + GSPMP	3.1503	0.3868
Res2Net50 + GhostVlad	3.5445	0.3899
Fusion	**2.4511**	**0.3036**

Table 5: *Finetune performance of each single model and the fused system on task3 dev*

Model	EER (%)	minDCF
ResNet34 + GSP	1.7951	0.2322
ResNet34 + GSPMP	2.2837	0.2596
ResNet34 + GhostVlad	2.0023	**0.2263**
Res2Net50 + GSP	**1.6373**	0.2661
Res2Net50 + GSPMP	1.6517	0.2303
Res2Net50 + GhostVlad	1.8718	0.2525
Fusion	**1.4273**	**0.1878**

In the end, the final result from the fusion system is submitted and evaluation trial results (30% of test set) on task1, task2 and task3 are shown in Table 6.

Table 6: *Finetune performance of the final fusion system on all three tasks on leaderboards*

Tasks	EER (%)	minDCF
task1	3.03	0.3152
task2	3.03	0.3632
task3	3.06	0.2849

The results show that multi-channel ResNet and Res2Net are promising backbone models by taking advantage of multi-channel information.

5. Conclusions

The report presents the system submitted to the Far-Field Speaker Verification Challenge 2020. In this system multi-channel ResNet and Res2Net are used as backbone model, data augmentation like adding noise, room acoustic simulating, speed perturbation and SpecAugment are used in both pre-training and fine-tune stages. Six models is fused with refined score-weighting to get the state of art performance in far-field scenario. Due to time constraints, we don’t try more. New data augmentation methods and better fusion method might achieve better results in the future.

6. References

[1] Qin, Xiaoyi, Ming Li, Hui Bu, Wei Rao, Rohan Kumar Das, Shrikanth Narayanan, and Haizhou Li. "The INTERSPEECH 2020 Far-Field Speaker Verification Challenge." arXiv preprint arXiv:2005.08046 (2020).

[2] Qin, Xiaoyi, Ming Li, Hui Bu, Rohan Kumar Das, Wei Rao, Shrikanth Narayanan, and Haizhou Li. "The FFSVC 2020 Evaluation Plan." arXiv preprint arXiv:2002.00387 (2020).
[3] Cai, Danwei, Xiaoyi Qin, and Ming Li. "Multi-Channel Training for End-to-End Speaker Recognition Under Reverberant and Noisy Environment." In INTERSPEECH, pp. 4365-4369. 2019.
[4] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.
[5] Gao, Shanghua, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip HS Torr. "Res2net: A new multi-scale backbone architecture." IEEE transactions on pattern analysis and machine intelligence (2019).
[6] Wan, Li, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. "Generalized end-to-end loss for speaker verification." In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4879-4883. IEEE, 2018.
[7] Scheibler, Robin, Eric Bezzam, and Ivan Dokmanić. "Pyroomacoustics: A python package for audio room simulation and array processing algorithms." In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 351-355. IEEE, 2018.
[8] Snyder, David, Guoguo Chen, and Daniel Povey. "Musan: A music, speech, and noise corpus." arXiv preprint arXiv:1510.08484 (2015).
[9] Park, Daniel S., William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. "Specaugment: A simple data augmentation method for automatic speech recognition." arXiv preprint arXiv:1904.08779 (2019).
[10] Yamamoto, Hitoshi, Kong Aik Lee, Koji Okabe, and Takafumi Koshinaka. "Speaker Augmentation and Bandwidth Extension for Deep Speaker Embedding." In INTERSPEECH, pp. 406-410. 2019.
[11] Xie, Weidi, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. "Utterance-level aggregation for speaker recognition in the wild." In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5791-5795. IEEE, 2019.
[12] Liu, Weiyang, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. "Sphereface: Deep hypersphere embedding for face recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 212-220. 2017.