The SJTU X-LANCE Lab System for CNSRC 2022
Zhengyang Chen, Bei Liu, Bing Han, Leping Zhang, Yanmin Qian
MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{zhengyang.chen, beiliu, hanbing97, zhangleying, yannin@sjtu.edu.cn}

Abstract

This technical report describes the SJTU X-LANCE Lab system for the three tracks in CNSRC 2022. In this challenge, we explored the speaker embedding modeling ability of deep ResNet (Deeper r-vector). All the systems are only trained on the Cnceleb training set and we use the same systems for the three tracks in CNSRC 2022. In this challenge, our system ranks the first place in the fixed track of speaker verification task. Our best single system and fusion system achieve 0.3164 and 0.2975 minDCF respectively. Besides, we submit the result of ResNet221 to the speaker retrieval track and achieve 0.4626 mAP. More importantly, we have helped the wespeaker [1] toolkit reproduce our result: https://github.com/wenet-e2e/wespeaker.

1. Data Usage
For all three tracks in the CNSRC 2022, we follow the same data usage setup. 797 speakers from CN-Celeb1 dev [2] and 1996 speaker from CN-Celeb2 [3] are used as the training data.

1.1. Data Processing
As shown in [3], there are many short utterances less than 2s in CN-Celeb dataset. In our experiment, we first concatenate the short utterances from the same genre and same speaker to make them longer than 5s. It should be noted that we only do this operation on the training set. After doing this operation, the training utterance number is reduced from 632,740 to 508,228.

1.2. Data Augmentation
In our experiments, three different augmentations are applied:

- Additive Noise: We use the audios from MUSAN [4] as the additive noise in our experiment to do data augmentation.
- Reverberation: The impulse response from RIR\(^1\) is used to do reverberation data augmentation.
- Speed Perturbation: We randomly speech up or slow down an utterance with ratio 1.1 and 0.9 to do speed perturbation. The utterance with a new speed will be considered from a new speaker [5, 6, 7].

We do all the data augmentations online. For each utterance in the training process, we independently decide whether to do each data augmentation with a probability 0.6.

\(^1\)https://www.openslr.org/28/

1.3. Acoustic Feature Extraction and Processing
In our experiment, we extract the 80-dimensional fbank feature for each utterance and then do the utterance-wise mean normalization along the time dimension. We did not apply voice activity detection (VAD) in our experiment.

2. System Architecture

2.1. Revisiting r-vector
r-vector is introduced in [8] and is the winning system of VoxSRC 2019 [9]. Although different ResNet variants [10, 11, 12] are proposed for the speaker embedding learning task, they differ in various aspects such as the convolutional type (1D vs. 2D), network width, kernel size, pooling methods, etc. In our experiences, we found the r-vector [8, 13] is the most stable one considering the performance and scalability. The 34-layer r-vector architecture used in this work is shown in Table 1, which is nearly the same as the original r-vector, while we use 80 dimensional Fbank instead of the 40 dimensional one. The ResNet backbone is used to transform the fbank feature into deep feature representations. Then, statistic pooling [14] is used to map the variable-length feature sequence to fixed-dimensional representation.

| Layer name     | Structure                  | Output          |
|----------------|----------------------------|-----------------|
| Input          | 80 × Frame Num × 1         | 80 × Frame Num × 1 |
| Conv2D-1       | 3 × 3, Stride 1            | 80 × Frame Num × 32 |
| ResNetBlock-1  | 3 × 3, 32 × 3, Stride 1    | 80 × Frame Num × 32 |
| ResNetBlock-2  | 3 × 3, 64 × 4, Stride 2    | 40 × Frame Num / 2 × 64 |
| ResNetBlock-3  | 3 × 3, 128 × 8, Stride 2   | 20 × Frame Num / 8 × 128 |
| ResNetBlock-4  | 3 × 3, 256 × 3, Stride 2   | 10 × Frame Num / 8 × 256 |
| StatsPooling   | 20 × 256                   | 5120            |
| Flatten        |                            | 256             |

1.4. Make r-vector deeper
It has been shown in [15] that impressive improvement can be achieved via deepening the speaker embedding learner, to make the r-vector more powerful, we extend it to a deeper version. The general configuration for these “deeper r-vector” is shown in Table 2.

We can change parameters \((N_1, N_2, N_3, N_4)\) in Table 2 to get deep ResNet with different layers and we list the configu-
ration for the deep ResNet used in our experiment in Table 3. Besides, we also use the DF-ResNet in [15] and make it deeper to 287 layers.

Table 2: Model architecture for deep ResNet.

| Layer name   | Structure                      | Output                          |
|--------------|--------------------------------|---------------------------------|
| Input        |                                | 80 x Frame Num x 1              |
| Conv2D-1     | 3 × 1, Stride 1                |                                |
| ResNetBlock-1| 1 × 1, 32, 3 × 3, 3 × 3, 1 × 1, 128 | 80 x Frame Num x 128           |
| ResNetBlock-2| 1 × 1, 64, 3 × 3, 1 × 1, 256    | 40 x Frame Num x 2 × 256        |
| ResNetBlock-3| 1 × 1, 128, 3 × 3, 1 × 1, 512   | 20 x Frame Num x 4 × 512        |
| ResNetBlock-4| 1 × 1, 256, 3 × 3, 1 × 1, 1024  | 10 x Frame Num x 8 × 1024       |
| StatisticPooling |                                |                                |
| Flatten      |                                | 20480                           |
| Emb Layer    |                                | 256                             |

Table 3: Configuration for different deep ResNet.

| Deep ResNet Name | (N1, N2, N3, N4) |
|------------------|------------------|
| ResNet132        | (3, 8, 36, 3)    |
| ResNet221        | (6, 16, 48, 3)   |
| ResNet293        | (10, 20, 64, 3)  |

3. System Training

In our experiment, we split the training process into two stages. In the first stage, we do the general classification training. In the second stage, we do the large margin finetuning, which is first proposed in [16]. At each stage, SGD is used as the optimizer to update our models. The learning rate is exponentially decreased from an initial value to a final value.

3.1. Stage I

In stage I, we use the additive angular margin (AAM) [17, 18] loss as the training objective. As mentioned in section 1.2, after applying speed perturbation, the speaker number is changed from 2793 to 8379. The scale and margin in the AAM loss are set to 32 and 0.2 respectively. We trained all the systems for 165 epochs. In each epoch, we go through the whole training set and randomly sample 2s segment from each utterance to build the training batch. In this stage, the initial value and final value of the learning rate are set to 0.1 and 0.00005 respectively.

3.2. Stage II: Large Margin Finetuning

In stage II, we use the large margin finetuning [16] based on the model from stage I. In this stage, we abandon the speed perturbation augmentation and the speaker classification number is 2793. All the systems are trained for another 5 epochs. Besides, we randomly sampled 6s segments from each utterance to construct a training batch and change the margin in AAM to 0.5. In this stage, the initial value and final value of the learning rate are set to 0.0001 and 0.000025 respectively.

4. System Scoring

4.1. Scoring for Speaker Verification

In our experiment, we use cosine similarity as the scoring method. Besides, we also add the adaptive score normalization [19] and we set the impostor cohort size to 600. The impostor cohort is estimated from the training set by average the embeddings for each training speaker. In the cneceleb evaluation trial, there are multiple utterances for each enrollment speaker. We tried three different ways to leverage the multiple utterances during the scoring for each trial pair.

Utt-Concat: In this strategy, we concatenate multiple utterances, which belong to the same enrollment speaker, to one long utterance. Then, we can directly get the score between enrollment utterance and test utterance using the scoring method mentioned above.

Emb-Avg: In this strategy, we first extract the speaker embedding for each utterance. Then, we average the embeddings belonging to the same enrollment speaker to get enrollment embedding.

Score-Avg: In this strategy, for each utterance from the same enrollment speaker, we compute its score with the test utterance. Then, the scores are averaged to be used as the score between the enrollment speaker and the test utterance.

4.2. Scoring for Speaker Retrieval

In the speaker retrieval task, we score each enrollment utterance against all the utterances in the large data pool. The scoring method is the same as the method used in the speaker verification task. Then, the utterances in the data pool with top 10 scores are considered as the retrieval results.

5. Results

5.1. Speaker Verification

5.1.1. Results comparison between different scoring strategies.

In this section, we will compare different ways to score the speaker verification system. The corresponding results are listed in Table 4. The results show that applying asnorm after cosine scoring can consistently bring further improvement. Actually, we also did the score calibration [16] in our experiment. However, the score calibration can only improve the EER but degrade the minDCF and we abandon it finally. Besides, we also compare the different ways to leverage the multiple utterances belonging to one enrollment speaker and we have given a detailed description of these methods in section 4.1. From the results, we find the Emb-Avg achieves the best result on the Cneceleb evaluation trial. The asnorm and Emb-Avg are applied to all the systems in the following sections.

Table 4: Results comparison between different scoring methods and different strategies to combine multiple utterances within one enrollment speaker. The results are from the ResNet34 model after stage I training.

| Scoring Method    | Enroll Comb | minDCF (0.01) | EER (%) |
|-------------------|-------------|---------------|---------|
| Cosine            | Utt-Concat  | 0.4391        | 7.305   |
| Cosine            | Emb-Avg     | 0.4004        | 6.922   |
| Cosine + ASnorm   | Utt-Concat  | 0.4035        | 7.085   |
| Cosine + ASnorm   | Emb-Avg     | 0.3707        | 6.590   |
| Cosine + ASnorm   | Score-Avg   | 0.4419        | 6.739   |
5.1.2. Results for different systems

Here, we listed results for all the systems in Table 5. As we expected, the deep ResNet has a strong modeling ability and can avoid overfitting problem at the same time. For the submission of the speaker verification task, we fuse all the systems with large margin finetuning based on their performance. Besides, we also list the false negative rate (FNR) and false positive rate (FPR) in Table 5. We get these two values by setting the score threshold to the same threshold value when we get minDCF (0.01). The results in Table 5 show that the minDCF is positively correlated with FNR and we can interpret the minDCF (0.01) as how the system performs on FNR in the case of very low FPR.

5.2. Speaker Retrieval

For the retrieval, we use the ResNet221+LM system in Table 5 to score each enrollment utterance against all the test utterances. Then we get the results with top 10 scores following the strategy described in section 4.2.

| System          | Params # | minDCF (0.01) | EER (%) | FNR (%) | FPR (%) |
|-----------------|----------|---------------|---------|---------|---------|
| ResNet34        | 6.63M    | 0.3958        | 7.981   | 35.29   | 0.043   |
| ResNet34        | 6.63M    | 0.3707        | 6.590   | 31.73   | 0.054   |
| ResNet152       | 19.8M    | 0.3386        | 5.762   | 29.34   | 0.045   |
| ResNet221       | 23.8M    | 0.3270        | 5.543   | 28.08   | 0.046   |
| ResNet293       | 28.6M    | 0.3202        | 5.553   | 27.92   | 0.041   |
| DF-ResNet       | 14.8M    | 0.3361        | 6.279   | 28.83   | 0.048   |
| ResNet34 + LM   | 6.63M    | 0.3543        | 6.221   | 30.06   | 0.054   |
| ResNet152 + LM  | 19.8M    | 0.3251        | 5.452   | 28.66   | 0.039   |
| ResNet221 + LM  | 23.8M    | 0.3179        | 5.284   | 28.27   | **0.035** |
| ResNet293 + LM  | 28.6M    | 0.3164        | 5.227   | 27.82   | 0.038   |
| DF-ResNet + LM  | 14.8M    | 0.3185        | 6.117   | 27.46   | 0.044   |
| Fusion          | -        | 0.2975        | 4.911   | 25.28   | 0.045   |

We didn’t apply speed perturbation on ResNet34 * as a comparison.

5. References

[1] Hongji Wang, Chengdong Liang, Shuai Wang, Zhengyang Chen, Binbin Zhang, Xu Xiang, Yunlei Deng, and Yanmin Qian, “Wespeaker: A research and production oriented speaker embedding learning toolkit,” in ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023, pp. 1–5.

[2] Yue Fan, JW Kang, LT Li, KC Li, HL Chen, ST Cheng, PY Zhang, ZY Zhou, YQ Cai, and Dong Wang, “CN-Celeb: a challenging chinese speaker recognition dataset,” in ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 7604–7608.

[3] Lantian Li, Ruqi Li, Jiawen Kang, Yue Fan, Hao Cui, Yunqi Cai, Ravichander Vipperla, Thomas Fang Zheng, and Dong Wang, “CN-Celeb: multi-genre speaker recognition,” Speech Communication, 2022.

[4] David Snyder, Guoguo Chen, and Daniel Povey, “MUSAN: A Music, Speech, and Noise Corpus,” 2015, arXiv:1510.08484v1.

[5] Hitoshi Yamamoto, Kong Aik Lee, Koji Okabe, and Taku-fumi Koshinaka, “Speaker augmentation and bandwidth extension for deep speaker embedding,” in Interspeech, 2019, pp. 406–410.

[6] Weiqing Wang, Danwei Cai, Xiaoyi Qin, and Ming Li, “The dku-dukeece systems for voxceleb speaker recognition challenge 2020,” arXiv preprint arXiv:2010.12731, 2020.

[7] Miaozhao, Yufeng Ma, Min Liu, and Minqiang Xu, “The speakin system for voxceleb speaker recognition challenge 2021,” arXiv preprint arXiv:2109.01989, 2021.

[8] Hossein Zeinali, Shuai Wang, Anna Silnova, Pavel Matějka, and Oldřich Plchot, “But system description to voxceleb speaker recognition challenge 2019,” arXiv preprint arXiv:1910.12592, 2019.

[9] Joon Son Chung, Arsha Nagrani, Ernesto Coto, Weidi Xie, Mitchell McLaren, Douglas A Reynolds, and Andrew Zisserman, “Voxsrc 2019: The first voxceleb speaker recognition challenge,” arXiv preprint arXiv:1912.02522, 2019.

[10] Weidi Xie, 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). IEEE, 2019, pp. 5791–5795.

[11] Weicheng Cai, Jinkun Chen, and Ming Li, “Data augmentation using deep generative models for embedding based speaker recognition,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 2598–2609, 2020.
[14] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, “Deep neural network embeddings for text-independent speaker verification,” in *Interspeech*, 2017, pp. 999–1003.

[15] Bei Liu, Zhengyang Chen, Shuai Wang, Haoyu Wang, Bing Han, and Yanmin Qian, “DF-ResNet: Boosting Speaker Verification Performance with Depth-First Design,” in *Proc. Interspeech 2022*, 2022, pp. 296–300.

[16] Jenthe Thienpondt, Brecht Desplanques, and Kris De-muyck, “The idlab voxceleb speaker recognition challenge 2020 system description,” *arXiv preprint arXiv:2010.12468*, 2020.

[17] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[18] Xu Xiang, Shuai Wang, Houjun Huang, Yanmin Qian, and Kai Yu, “Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition,” in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 1652–1656.

[19] Sandro Cumani, Pier Domenico Batzu, Daniele Colibro, Claudio Vair, Pietro Laface, and Vasilios Vasilakakis, “Comparison of speaker recognition approaches for real applications,” in *Twelfth annual conference of the international speech communication association*, 2011.