HCCL&Tencent system for CN-Celeb Speaker Recognition Challenge 2022

Zhenduo Zhao[1,2,3], *Shengyu Yao*[3], *Zhuo Li*[1,2], *Yiqian Pan*[3], *Wenchao Wang*[1]

[1]Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences
[3]Tencent inc. Beijing

{zhaozhenduo,lizhuo,wangwenchao}@hccl.ioa.ac.cn
{shengyuyao,irenepan}@tencent.com

Abstract

This report describes our submission to the task1 open track and task2 open track for CN-Celeb Speaker Recognition Challenge 2022(CNSRC 2022). Our best system achieves minDCF 0.3121 and EER 5.1140 in task1 open track, mAP 0.7071 in task2 open track.

Index Terms: speaker verification, speaker retrieval, cn-src2022

1. System Description

1.1. Data

Our trainset consists of data from CN-Celeb-T [1] and Vox-Celeb-1&2 [2, 3] and a private corpus. CN-Celeb-T includes 632740 records from 2793 speakers. We filtered out 1327 speakers from CN-Celeb-T based on the number of files and duration, named CN-Celeb-T-Filtered1, then we got 2603 speakers by relaxing the threshold, named CN-Celeb-T-Filtered2. Vox-Celeb-1&2 include 1281762 records from 7245 speakers. Our private corpus has 16003 records from 442 speakers collected from the internet and is independent of CN-Celeb. There are 9014 speakers in total from the original dataset. As speaker recognition systems benefit from more speaker numbers, we first introduce speed perturbation [4] of 0.9 and 1.1 to generate extra twice speakers using SoX. Then we have 24540 speakers and 4209420 utterances as we did not use the speed perturbation of Vox-Celeb-1. Data augmentation was performed in online mode with MUSAN [5] and RIRS [6]. Finally, we use log Mel-Filter Bank Enegyies (FBank) with 80 bins.

For evaluation, the task1 test set has 196 enrollment from 196 speakers and 17777 tests. Task2 development set has 5 speakers with 10 test utterances each and 20,000 non-target utterances; the evaluation set has 25 speakers with 10 test utterances each and 500,000 non-target utterances.

1.2. Model Architecture

Convolutional Neural Networks are currently the most popular network structure. The model we used is a modified version of ResNetSE [7, 8], similar to our previous work in VoxSRC2021 [9, 10]. It contains 4 residual blocks, and the structure of the basic block in each residual block is shown in figure 1. Residual block 1&2 uses BasicBlock with SE, and residual block 3&4 uses BasicBlock. After residual blocks, we use channel-wise correlation pooling [11] to aggregate frame-level features. The feature map was then transformed into 512-dimensional embeddings.

Table 1: *Dataset Summary.*

dataset	speakers	utterances
CN-Celeb-T	2793	732740
CN-Celeb-T-Filtered1	1327	246698
CN-Celeb-T-Filtered2	2603	607378
Vox-Celeb-1	1251	153516
Vox-Celeb-2	5994	1128246
private corpus	442	16003
CN-Celeb-E	196	17777
Task2-dev	5	50 in 20050
Task2-eval	25	250 in 500250

1.3. Loss Funcsion

Models are trained with AAM-Softmax Loss [12]. It is formulated as:

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

where m is the margin, s is the scale factor, N is the number of speaker.

To further improve the performance of our system, we introduce subcenter [13] and intertop-k [14] to the loss function.

Subcenter split every class in the training dataset into k subclasses and chooses the largest logits when computing losses. It helps the model filter out noise samples.

Intertop-k adds a margin in the k-closest classes and thus forces the model to be more discriminating. It can be considered as a hard sample mining method. AAM-Softmax with intertop-k is formulated as:

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

where $\phi(\theta_j)$ is:

$$\phi(\theta_j) = \begin{cases} cos(\theta_j+m), & j \in \arg\underset{1\leq n\leq N}{topK}(cos(\theta_j)) \\ cos(\theta_j). & Others \end{cases} \tag{3}$$

Intertop-k operation comes after the subcenter in our implementation.

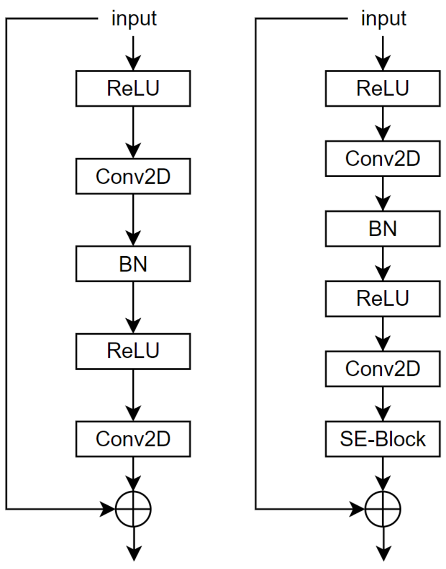

Figure 1: Residual block of ResNetSE. Left: Basic Block, right: SE Basic Block.

Table 2: *Model Architecture.S:stride, L:length of input.*

Layer	Kernel size	Output
Conv1	3x3x32,S=1	1x1
Block1	$\begin{bmatrix} 3\times3\times32 \\ 3\times3\times32 \\ SELayer \end{bmatrix} \times 6, S=1$	$L\times80\times32$
Block2	$\begin{bmatrix} 3\times3\times64 \\ 3\times3\times64 \\ SELayer \end{bmatrix} \times 16, S=2$	$L/2\times40\times64$
Block3	$\begin{bmatrix} 3\times3\times128 \\ 3\times3\times128 \end{bmatrix} \times 24, S=2$	$L/4\times20\times128$
Block4	$\begin{bmatrix} 3\times3\times256 \\ 3\times3\times256 \end{bmatrix} \times 3, S=2$	$L/8\times10\times256$
Pooling	-	10080
Linear	10080	512

1.4. Backend

This section describes our scoring backend.

1.4.1. *Task1*

We adopted 4sx10 cosine scoring. We expanded each utterance if its duration was less than 10s, evenly took 4s segments, and then extracted embeddings of pieces. For each trial, the 10 parts of enroll and test utterance were scored in pairs to get a 10x10 similarity matrix, then took the mean of the similarity matrix as the score of the trial. The formulation is given by:

$$score(x_1, x_2) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} cos(x_{1_m}, x_{2_n}), \quad (4)$$

$$cos(a,b) = \frac{a^T b}{||a||||b||}, \quad (5)$$

where x_1, x_2 is a pair of utterances, $M = N = 10$ is the number of split from x_1, x_2, x_{1_m} is the embedding of mth chunk of utterance x_1 and x_{2_n} is the embedding of nth chunk of utterance x_2.

As for score normalization, we extract embeddings of cn-celeb-T with full length and then average speaker-wise to generate a cohort of 2793 speakers. Then the score is calibrated using adaptive scoring normalization(AS-Norm) [15] with top k imposter scores, which is formulated as:

$$\hat{\mathcal{S}}(x_1, x_2) = \frac{\mathcal{S}(x_1, x_2) - \mu_1}{\sigma_1} + \frac{\mathcal{S}(x_1, x_2) - \mu_2}{\sigma_2}, \quad (6)$$

Here we set $k = 500$ for track1.

1.4.2. *Task2*

Task 2 and Task 1 share the same scoring backend, in addition to that, we have added a music calibration module after as-norm.

We found the score between two speech records was usually higher than that between speech and singing records. To compensate for the mismatch, we made use of the existing music detection system to identify whether the test utterance was a singing record or not. After AS-Norm, we added a bias to trials in which test embedding was a singing record. Mobile Net V2 [16] was used to train the singing classification model. The training data consisted of 640 hours of private speech data, 490 hours of songs, and 180 hours of pure music. Songs and music were collected from the internet.

We used the average of every single system to get a fused score. We computed the average score of single system results, then performed music calibration, and finally took the top-10 of each target speaker.

1.5. Training Protocal

We trained models with a two-stage protocol. Experiments were built on the top of PyTorch [17].

In the first stage, Adam [18] optimizer with weight decay of 5e-5 was used. Learning rate followed cyclic learning rate [19] with the same settings as ECAPA-TDNN [20], where learning rate linear increased from 1e-8 to 1e-3 for 650k steps and linearly decreased back to 1e-8 for 650k steps in one cycle, repeat for 4 cycles. We use 2 GPUs with 400 mini-batch. 300 frames of each sample in one batch are adopted to avoid over-fitting. The margin is 0.2, and the scale is 30 during the training stage.

In the second stage, we only use a subset of the train set, expand frame length from 300 to 600, increase the margin of aam-softmax from 0.2 to 0.5, decrease the peak learning rate of the cyclic schedule from 1e-3 to 1e-4, narrow the cycle range from 130k to 20k, and only train 1 cycle as we found that the model rapidly overfit and the performance decayed dramatically. Sub-center and intertop-k were removed from the loss function by default.

We trained several models based on ResNetSE101 but with different configurations.

S1: In the first stage, model was trained with subcenter k=3, and finetuned with 2793 speakers in CN-Celeb-T.

S2: In the first stage, model was trained with subcenter k=3, intertop-k k=5, m=0.1, and finetuned with 2603 speakers in CN-Celeb-T.

S3: In the first stage, model was trained with subcenter k=3, intertop-k k=5, m=0.1, and finetuned with 2603 speakers in CN-

Celeb-T and the weight decay was set to 0.

S4: In the first stage, model was trained with subcenter k=3, intertop-k k=5, m=0.06, and finetuned with 3222 speakers in CN-Celeb-T and private corpus.

S5: In the first stage, model was trained with subcenter k=3, intertop-k k=5, m=0.06. In the second stage, model was finetuned with 3222 speakers in CN-Celeb-T and private corpus, and intertop-k was preserved.

2. Results

This section shows our results on task1 open track and task2 open track.

2.1. Task1 submission

Table 3 shows the result on the task1 open track. Our best single system achieved EER 5.11 and minDCF 0.2920, while the submission to the leader board achieved EER 5.1140, and minDCF 0.3121 because we did not update our result. As we focused more on task2, we did not try to fuse them. Here we only list single system performance.

Table 3: *Results on Task1 open track*

model	EER	minDCF(0.01)
S1	5.31	0.2973
S2	5.39	0.2886
S3	5.00	0.2925
S4	5.47	0.3063
S5	5.11	0.2920

2.2. Task2 submission

Table 4 shows the result of the system for the Task2 open track.System S1 got mAP=0.5507 before as-norm, and after as-norm(k=500), it achieved mAP=0.6647. We found out the optimal k in task2 was around $k = 10$, and the score was mAP=0.6790 by S1. Music calibration did not bring performance improvement on the development set because only 3 target records are recognized as music. We found that the mAP curve will maintain the performance within a certain range after music calibration; this constant interval is highly consistent on the mAP curves across different systems; we then selected the midpoint as the threshold for evaluation. Although we used the same model backbone, we found system fusion still works. On the development set, the best performance was 0.920 with 2 systems fused. However, with 5 systems, we achieved mAP=0.7071 on the evaluation set, which is the second place on the leader board.

3. Conclusion

This report describes our submission to CNSRC-2022 task1 open track and task2 open track in detail.

Table 4: *Results on task2 open track. MC: music calibration*

model	dev	dev(MC)	eval	eval(MC)
S1	0.915	0.915	0.6661	0.6790
S2	0.911	0.911	–	–
S3	0.900	0.900	–	–
S4	0.911	0.911	–	–
S5	0.915	0.915	–	–
fusion				
S1-2	0.920	0.920	–	0.6874
S1-3	0.917	0.917	–	0.7044
S1-4	0.915	0.915	–	0.7068
S1-5	0.915	0.915	–	0.7071

4. References

[1] Yue Fan, Jian Kang, Lingjun Li, Kan Li, Haolin Chen, Sitong Cheng, Peng Zhang, Ziya Zhou, Yunqi Cai, and Dong Wang, "Cn-celeb: a challenging chinese speaker recognition dataset," *international conference on acoustics, speech, and signal processing*, 2019.

[2] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: A large-scale speaker identification dataset," *conference of the international speech communication association*, 2017.

[3] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," *conference of the international speech communication association*, 2018.

[4] Hitoshi Yamamoto, Kong Aik Lee, Koji Okabe, and Takafumi Koshinaka, "Speaker augmentation and bandwidth extension for deep speaker embedding," *conference of the international speech communication association*, 2019.

[5] David Snyder, Guoguo Chen, and Daniel Povey, "Musan: A music, speech, and noise corpus," *arXiv: Sound*, 2015.

[6] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," *international conference on acoustics, speech, and signal processing*, 2017.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv: Computer Vision and Pattern Recognition*, 2015.

[8] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu, "Squeeze-and-excitation networks," *computer vision and pattern recognition*, 2018.

[9] A. Brown, Jaesung Huh, Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxsrc 2021: The third voxceleb speaker recognition challenge," 2021.

[10] Yao Shengyu, Fang Xiang, Yan Jie, Li Jingdong, and Pan Yiqian, "Sogou system for the voxceleb speaker recognition challenge 2021," *arXiv: Sound*, 2021.

[11] Themos Stafylakis, Johan Rohdin, and Lukás Burget, "Speaker embeddings by modeling channel-wise correlations," in *Interspeech*, 2021.

[12] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *arXiv: Computer Vision and Pattern Recognition*, 2018.

[13] Jiankang Deng, Jia Guo, Tongliang Liu, Mingming Gong, and Stefanos Zafeiriou, "Sub-center arcface: Boosting face recognition by large-scale noisy web faces," in *ECCV*. 2020, pp. 741–757, Springer.

[14] Miao Zhao, Yufeng Ma, Min Liu, and Minqiang Xu, "The speakin system for voxceleb speaker recognition challange 2021," *arXiv: Sound*, 2021.

[15] Sandro Cumani, Pier Domenico Batzu, Daniele Colibro, Claudio Vair, Pietro Laface, and Vasileios Vasilakakis, "Comparison of speaker recognition approaches for real applications," *conference of the international speech communication association*, 2011.

[16] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," *computer vision and pattern recognition*, 2018.

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, "Pytorch: An imperative style, high-performance deep learning library," *neural information processing systems*, 2019.

[18] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv: Learning*, 2014.

[19] Leslie N. Smith, "Cyclical learning rates for training neural networks," *workshop on applications of computer vision*, 2015.

[20] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.