THE VOLKSWAGEN-MOBVOI CN-CELEB SPEAKER RECOGNITION CHALLENGE 2022 SYSTEM DESCRIPTION

YingWei Tan, XueFeng Ding

Volkswagen-Mobvoi (Beijing) Information Technology Co., Ltd

{ywtan, xfding}@vw-mobvoi.com

ABSTRACT

Our submission to the track 1 of the CN-CELEB Speaker Recognition Challenge 2022 (CNSRC 2022) is described by this report. The track 1 task only uses the CN-Celeb training set for training/tuning the system. The objective of this task is to improve performance on the standard CN-Celeb evaluation set. Based on the state-of-the-art SEResnet speaker embedding network, we explore a novel network architecture with split-attention, called ResNeSt, and novel hybrid statistics pooling methods. Based on these techniques, we achieve significant improvement over the SEResnet baselines. Furthermore, in-domain data finetuning, attention back-end methods, speaker-wise adaptive score normalization (AS-Norm) and score calibration on duration efficiently improve the robustness. Finally, our system is a fusion of 23 models and achieves eleventh place in the track 1 of CNSRC 2022. The minDCF of our submission is 0.4159, and the corresponding EER is 7.333%.

1. DATA

1.1. Training data

According to the evaluation plan of CNSRC 2022, the track 1 task is a fixed training condition where the system should only be trained using the CN-Celeb training set [20]. It consists of 632740 utterances from 2793 speakers. It is forbidden to use any other public or private speech data for training.

1.2. Test data

The enrollment data consists of 196 utterances from 196 speakers. The test data consists of 17777 utterances from 200 speakers. 3484292 trials are sampled from the CN-Celeb test dataset with only 200 speakers. Each trial in the test set contains a test utterance and a target model. The enrollment data for target model consists of one utterance.

1.3. Data preparation

In our experiments, we make use of all allowed training data with speaker label as our training data, totally 2793 labeled speakers. It is also used for training back-end models such as PLDA and attention back-end models. We extracted 81-dimensional log Mel filter bank energies based on Kaldi [13]. The window size is 25 ms, and the frame shift is 10 ms. 200 frames of features were extracted with energy-based voice activation detection (VAD).

2. MODELS

In this section, we will introduce the embedding extractors, pooling methods, finetuning strategies and several post-processing methods used in our system. In our experiment, the embedding extractors are firstly trained on all the available data for task1 in a text-independent mode. Then, we fine-tune the pre-trained models using in-domain data. Finally, post-processing methods are used to further improve the system performance. Compared with a regular SEResnet baseline, we propose our improvement on the network, pooling and scoring in the subsequent sections.

2.1. Network Structures

2.1.1. Baseline SEResnet Models

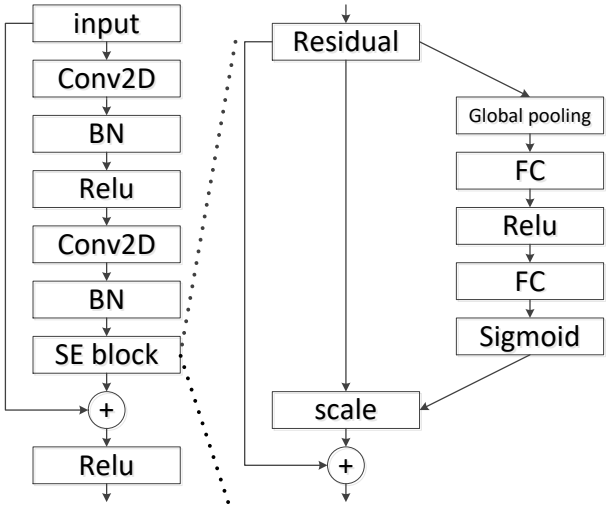

Figure 1. *the architecture of basic block in SEResnet model and the structure of the SE block.*

In speaker verification, the embedding extraction is mainly based on the Residual Neural Network (ResNet) architecture. ResNet is built on convolution layers. Unfortunately, the convolution layer does not exploit the dependencies between feature maps. For solving the problem, a channel attention mechanism called squeeze-and-excitation (SE), has recently been proposed in convolution layers and applied to speaker verification. We use ResNet with Squeeze-and-Excitation (SE) layer as our baseline model [4].

There are 4 basic blocks in a SEResnet model. Figure 1 depicts the architecture of basic block in SEResnet model and

the structure of the SE block. First, we produce a global information of each channel using a pooling layer. The pooling layer aggregates feature maps across their spatial dimension to a single numeric value. Thus, a vector of size n is obtained where n is equal to the number of feature maps. Then, the vector is introduced into a two-layer full connection neural network. A n dimensional output vector is obtained. These n values can now be used as weights on the original feature maps, scaling each channel based on its importance. The pooling layer plays a central role in the SE strategy.

The SE block can be simply integrated in CNN by inserting after the non-linearity following each convolution. In the case of ResNet, the classical integration strategy is to insert SE block after the final convolutional layer and before the skip connection branch. The idea to integrate the SE block before the skip connection branch, is to avoid noise in the skip connection branch and facilitate the learning of identity.

2.1.2. ResNeSt Models

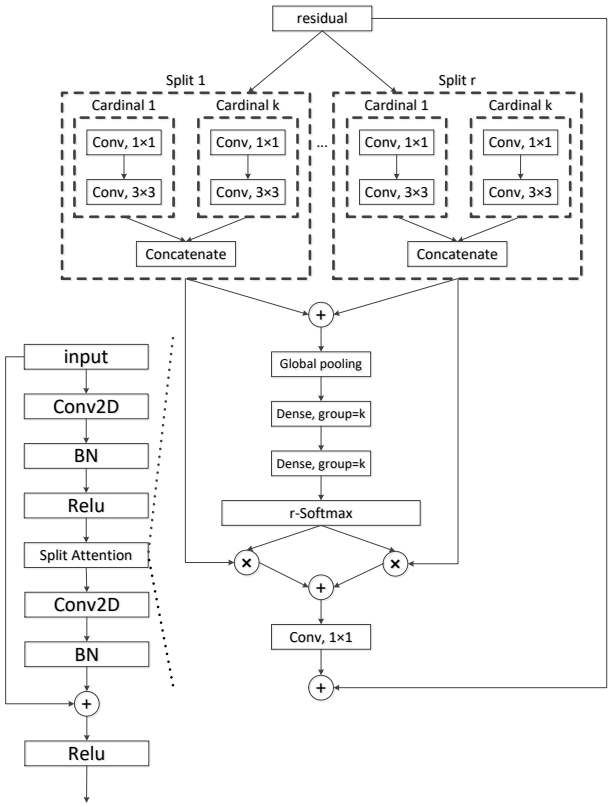

Figure 2. *the architecture of basic block in ResNeSt model and the implementation of a split-attention block in the basic block.*

The baseline SEResnet Model introduces a channel attention mechanism by adaptively recalibrating the channel feature responses. Inspired by the previous methods, ResNeSt network [14] integrates the channel-wise attention with multi-path network representation. The method captures cross-channel feature correlations, while preserving independent representation in the meta structure. A set of transformations is performed by a module of our network on low dimensional embeddings. The module concatenates their outputs as in a multi-path network. Such a computation block is called as a split-attention block. Stacking several split-attention blocks in ResNet style, we create a new ResNet variant which we refer to as split-attention network (ResNeSt).

Figure 2 give an overview of the split-attention block in ResNeSt models. The input featuremap is first divided into RK groups. In our experiments, R and K are set to 4 and 2. In this graph, the groups with same radix-index reside next to each other. Then, a summation is conducted across different splits, so that the featuremap groups with the same cardinality-index but different radix-index are fused together. A global pooling layer aggregates over the spatial dimension, while keeps the channel dimension separated, which is identical to conducting global pooling to each individual cardinal groups then concatenate the results. We can gather global contextual information with embedded channel-wise statistics with global average pooling across spatial dimensions. Then two consecutives fully connected (FC) layers with number of groups equal to cardinality are added after pooling layer to predict the attention weights for each split.

The idea of sequeeze-and-excitation is to employ a global context to predict channel-wise attention factors, first introduced in SE-net [15]. In this work, our method generalizes prior work of featuremap attention block within a group setting and remains computationally efficient.

2.2. Pooling Methods

2.2.1. Attentive statistics pooling

According to [5], attentive statistics pooling methods aim to capture the temporal information focusing on the importance of frames. In order to make the attention discriminate features from multiple aspects, an attention model calculates a scalar score e_t^i for each frame in a multi-resolution and multi-head way, as follows:

$$e_t^i = {(\mathrm{v}_i^T f(w_i h_t + b_i) + k_i)}\Big/{N_i}, \forall i \in \{1, \dots, I\} \qquad (1)$$

where $f(\cdot)$ is a non-linear activation function, such as tanh or ReLU, I is the number of attention heads, and N_i is temperature. Different heads have different temperature. The scores are normalized over all frames with a softmax function as follows:

$$\alpha_t^i = \frac{\exp(e_t^i)}{\sum_\tau^T \exp(e_\tau^i)} \qquad (2)$$

Note that the softmax is performed along the temporal axis.

2.2.2. Self-attentive pooling

In [9, 16], the authors used a more rigorous formulation based on a {value, key, query} tuple to construct the so-called self-attentive pooling mechanism.

Considering an input sequence $[h_1, h_2, \dots, h_T]$, where T is the length of the input sequence. The model transforms the input sequence into the query q^i as follows:

$$q^i = w_q^i g(h_t) \qquad (3)$$

where $g(\cdot)$ is statistics pooling function, and w_q^i is a trainable parameter.

As for key-value pairs, in order to reduce the number of model parameters, the input sequence $[h_1, h_2, \dots, h_T]$ is

directly assigned to the value sequence $[v_1, v_2, \dots, v_T]$ without any extra computation. The key vector k_t with d_k dimensions is obtained by a linear projection with a trainable parameter:

$$k_t = W_k h_t \tag{4}$$

where W_k is a trainable parameter. A scalar score is computed via scaled dot-product attention as:

$$u_t^i = \left(\frac{q^i \cdot k_t}{\sqrt{d_k}}\right) \Big/ N_i, \forall i \in \{1, \dots, I\} \tag{5}$$

The scores are normalized over all frames with a softmax function as follows:

$$\beta_t^i = \frac{\exp(u_t^i)}{\sum_\tau^T \exp(u_\tau^i)} \tag{6}$$

Note that the softmax is performed along the temporal axis.

2.2.3. Hybrid statistics pooling

By exploiting the advantage of two attention combination, we compute a hybrid weight:

$$\gamma_t^i = (\alpha_t^i + \beta_t^i)/2 \tag{7}$$

The i-th weighted mean vector μ^i can be calculated by summing up the element-wise products of each frame-level vector h_t and the hybrid weight γ_t^i:

$$\mu^i = \sum_{t=1}^T \gamma_t^i h_t \tag{8}$$

And the i-th weighted standard deviation vector σ^i can be acquired as follows:

$$\sigma^i = \sqrt{\sum_{t=1}^T \gamma_t^i h_t \odot h_t - \mu^i \odot \mu^i} \tag{9}$$

Finally, the utterance-level representation E of the hybrid statistics pooling layer is the vector concatenating μ^i and σ^i in all I attention heads:

$$E = [\mu^1; \dots; \mu^I; \sigma^1; \dots; \sigma^I] \tag{10}$$

2.3. Model Finetune

To further improve system performance on test data, we finetune the models with training data. In practice, we fix the parameter of the speaker network except loss functions and retrain the model. The AM-Softmax loss was replaced by AAM-Softmax loss. The margin is increased from 0.2 to 0.5.

2.4. Scoring

The trained networks are evaluated on the CN-Celeb test dataset. We compared three different scoring methods on the trained models. Firstly, we use a cosine similarity method to measure whether the two utterances are from the same speaker after the embeddings are extracted. Secondly, we use PLDA [10] for scoring. PLDA models are trained on labeled speakers after the speaker embeddings are extracted. Finally, to make better use of intra-relationships of the utterances, a novel attention back-end model [11] is applied. A balanced batch strategy is adopted to train the attention back-end model. For each mini-batch, assuming it has M speakers and K speaker embeddings per speaker, the size of one mini-batch is $M \times K$. In our experiment, M and K are set to 256 and 5.

2.5. Score calibration

According to [17], the speaker similarity score is largely affected by the quality of the trial speeches. Hence, the quality function mainly based on the duration of the test speech is applied. Assuming speech duration for enrollment is long enough, the rescoring method is as follows:

$$\hat{S} = S + C \cdot f(d_t) \tag{11}$$

where S is the raw score, C is a scaling constant, d_t is the duration (in seconds) of test speech in scoring trial, and f is the duration-based quality function:

$$f(d_t) = \frac{1}{d_t} \tag{12}$$

The optimal value of C is 0.21 or 1.2 in our final fusion systems.

3. RESULTS

The main performance metric adopted by CNSRC challenge is normalized minimum Detection Cost Function (MinDCF). Besides, the equal error rate (EER) is also a very important performance metric in speaker verification.

3.1. Compare of SEResnet and ResNeSt

Table 1. *Compare of SEResnet and ResNeSt*

Methods	CN-Celeb evaluation set	
	EER(%)	MinDCF(0.01)
SEResnet-34	10.96	0.5313
ResNeSt-34	10.15	0.5219
ResNeSt-50	**9.901**	0.5151
SEResnet-34+ attentive statistics pooling	10.54	0.5194
SEResnet-34+self-attentive statistics pooling	10.22	0.5193
SEResnet-34 +Hybrid statistics pooling	10.08	**0.5115**
ResNeSt-50 +Hybrid statistics pooling	10.17	0.5193

The performance of SEResnet model (baseline) and ResNeSt model is described in Table 1 respectively. It can be found that ResNeSt-34 performs better than SEResnet-34 under the same training strategy, improved by 7%/2% in EER/MinDCF compared with the former. After increasing the depth of the ResNeSt, ResNeSt-50 further improves the performance both in EER and MinDCF. Experiments indicate the importance of using ResNeSt-50. Additionally, by substituting for other statistics pooling [1, 5, 9] with hybrid statistics pooling, SEResnet-34 achieve substantial performance improvement. But no performance superposition was gained by ResNeSt-50.

3.2. Ablation study

In this subsection, we show our detailed ablation study on our ResNeSt-50 system. The ResNeSt-50 backbone is followed by statistics pooling and AM-Softmax. As Table 2 shows, our

ResNeSt-50 system's performance improved significantly on various trials by stacking our proposed methods gradually. First, we conducted our ablation studies by adding model finetune. The MinDCF was improved from 0.5151 to 0.5141. Using attention back-end models instead of cosine similarity methods, the EER further achieved 9.541%, and the MinDCF was 0.5044. The procedures above already boosted our ResNeSt-50 system's EER by relatively 3.64% and MinDCF by relatively 2.08%. Applying the speaker-wise AS-Norm further achieved 9.456% EER and 0.5026 MinDCF. The final score calibration process got 9.389% EER and 0.5014 minDCF. After completing the ablation study, our ResNeSt-50 system improved EER relatively 5.17% and minDCF relatively 2.66% in total.

Table 2. *Ablation Study on ResNeSt-50*

Methods	CN-Celeb evaluation set	
	EER(%)	MinDCF(0.01)
ResNeSt-50	9.901	0.5151
ResNeSt-50 + Model Finetune	9.901	0.5141
ResNeSt-50 + Model Finetune +attention back-end model	9.541	0.5044
ResNeSt-50 + Model Finetune + attention back-end model +asnorm	9.456	0.5026
ResNeSt-50 + Model Finetune + attention back-end model +asnorm + score calibration (C=0.21)	**9.389**	**0.5014**

3.3. Sub-Systems and Fusion Performance

All our sub-systems were described in Table 3. A total of 23 different styles were used to generate different representations. We found that a large model, such as ResNeSt-50, seemed to yield a better result compared to smaller models like our baseline system. ResNeSt-50 achieves the best performance both in EER and MinDCF by applying all these strategies, as it shows in system 13 and 14. It is worth mentioning that our ResNeSt-50 system with attention back-end models achieved a 0.5014 minDCF and 9.389% EER while our ResNeSt-50 system with cosine similarity methods achieved a 0.4979 minDCF and 9.518% EER. Based on the ResNeSt-50, system 15 to system 19 also obtained good results using different pooling methods and different scoring methods.

Comparing the results of system 1 to system 5, we can find that the novel network architecture with split-attention and hybrid statistics pooling methods improve the system performance respectively.

System 6 is a conventional ECAPA-TDNN model. System 7 is a standard Xi-Vector embedding method. System 8 is an efficient RESNET-34 system. They are built by ASV-Subtools [18]. System 9 and 10 are implemented by kaldi tools [13]. System 20 to system 23 are established by sunine tools, which has been published at https://gitlab.com/csltstu/sunine by Lantian Li et al. The MinDCF and EER of these systems are much worse compared to system 11 and 12, but they all contribute to the fusion system. Additionally, system 6 and 23 have the same model structure, but they are customized by different tools. System 10 and 20 are also like this. ResNet34L uses 16 base channels while ResNet34 uses 32 base channels. Additive noise from MUSAN corpus [21] and room impulse response (RIR) simulation [22] are used as data augmentation in system 10 and 19.

The fusion system is a score-level fusion using the bosaris toolkit [12]. The final fusion resulted in a 0.4159 minDCF and a 7.333% EER in the CNSRC 2022 challenge. The fusion result improved 21.72% relatively in minDCF and 33.09% relatively in EER compared to our SEResnet-34 model.

4. REFERENCES

1. 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.
2. 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.
3. 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.
4. Rouvier M, Bousquet P M. Studying squeeze-and-excitation used in CNN for speaker verification[J]. arXiv preprint arXiv:2109.05977, 2021.
5. Okabe K, Koshinaka T, Shinoda K. Attentive statistics pooling for deep speaker embedding[J]. arXiv preprint arXiv:1803.10963, 2018.
6. India M, Safari P, Hernando J. Self multi-head attention for speaker recognition[J]. arXiv preprint arXiv:1906.09890, 2019.
7. Wang Z, Yao K, Li X, et al. Multi-resolution multi-head attention in deep speaker embedding[C]//ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 6464-6468.
8. Lee K A, Wang Q, Koshinaka T. Xi-vector embedding for speaker recognition[J]. IEEE Signal Processing Letters, 2021, 28: 1385-1389.
9. Liu Y, He L, Liu W, et al. Exploring a unified attention-based pooling framework for speaker verification[C]//2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP). IEEE, 2018: 200-204.
10. Ioffe S. Probabilistic linear discriminant analysis[C]//European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2006: 531-542.
11. Zeng C, Wang X, Cooper E, et al. Attention back-end for automatic speaker verification with multiple enrollment utterances[J]. arXiv preprint arXiv:2104.01541, 2021.
12. Brümmer N, De Villiers E. The bosaris toolkit: Theory, algorithms and code for surviving the new dcf[J]. arXiv preprint arXiv:1304.2865, 2013.

13. Povey D, Ghoshal A, Boulianne G, et al. The Kaldi speech recognition toolkit[C]//IEEE 2011 workshop on automatic speech recognition and understanding. IEEE Signal Processing Society, 2011 (CONF).
14. Zhang H, Wu C, Zhang Z, et al. Resnest: Split-attention networks[J]. arXiv preprint arXiv:2004.08955, 2020.
15. Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.
16. Zhu H, Lee K A, Li H. Serialized Multi-Layer Multi-Head Attention for Neural Speaker Embedding[J]. arXiv preprint arXiv:2107.06493, 2021.
17. Jie Yan, Shengyu Yao, Yiqian Pan,Wei Chen, " The Sogou System for Short-duration Speaker Verification Challenge 2021", Interspeech 2021, Brno, Czechia.
18. Tong F, Zhao M, Zhou J, et al. ASV-Subtools: Open source toolkit for automatic speaker verification[C]//ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021: 6184-6188.
19. Dehak N, Kenny P J, Dehak R, et al. Front-end factor analysis for speaker verification[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2010, 19(4): 788-798.
20. Li L, Liu R, Kang J, et al. CN-Celeb: multi-genre speaker recognition[J]. Speech Communication, 2022.
21. Snyder D, Chen G, Povey D. Musan: A music, speech, and noise corpus[J]. arXiv preprint arXiv:1510.08484, 2015.
22. Ko T, Peddinti V, Povey D, et al. A study on data augmentation of reverberant speech for robust speech recognition[C]//2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017: 5220-5224.

Table 3 The EER(%) and MinDCF(0.01) results of our subsystems on the CN-Celeb evaluation set and the final result of our fusion system

System	Model	Pooling	Model Finetune	Scoring Method	asnorm	score calibration	CN-Celeb evaluation set	
							EER(%)	MinDCF(0.01)
1	SEResnet-34	statistics pooling	no	cosine	no	no	10.96	0.5313
2	SEResnet-34	multi-resolution [7]	no	PLDA	no	no	10.28	0.5195
3	SEResnet-34	hybrid statistics pooling	no	cosine	no	no	10.08	0.5115
4	ResNeSt-34	statistics pooling	no	cosine	no	no	10.15	0.5219
5	ResNeSt-34	statistics pooling	no	PLDA	no	no	10.09	0.5331
6	ECAPA-TDNN [3]	Attentive Stat Pooling	no	cosine	no	no	12.65	0.6010
7	TDNN-Xi-Vector [8]	Xi-Vector pooling	no	cosine	no	no	12.84	0.6266
8	RESNET-34 [2, 18]	statistics pooling	no	cosine	no	no	13.03	0.6170
9	i-vector [13, 19]	-	no	PLDA	no	no	13.87	0.6307
10	x-vector [1, 13]	statistics pooling	no	PLDA	no	no	12.19	0.5957
11	ResNeSt-50	statistics pooling	yes	attention back-end	no	no	9.541	0.5044
12	ResNeSt-50	statistics pooling	yes	attention back-end	yes	no	9.456	0.5026
13	ResNeSt-50	statistics pooling	yes	attention back-end	yes	yes, (C=0.21)	**9.389**	**0.5014**
14	ResNeSt-50	statistics pooling	no	cosine	yes	yes, (C=1.2)	**9.518**	**0.4979**
15	ResNeSt-50	multi-head [6] pooling	no	cosine	no	no	10.48	0.5283
16	ResNeSt-50	multi-head pooling	no	PLDA	no	no	9.969	0.5232
17	ResNeSt-50	Attentive Stat Pooling	no	cosine	no	no	9.941	0.5239
18	ResNeSt-50	Attentive Stat Pooling	no	PLDA	no	no	9.716	0.5051
19	ResNeSt-50	hybrid statistics pooling	no	PLDA	no	no	9.541	0.4742
20	TDNN [1]	statistics pooling	no	cosine	no	no	15.83	0.7276
21	ResNet34L	statistics pooling	no	cosine	no	no	11.87	0.5952
22	ResNet34	Attentive Stat Pooling	no	cosine	no	no	10.62	0.5494
23	ECAPA-TDNN	Attentive Stat Pooling	no	cosine	no	no	11.41	0.5977
fusion	1~23	-	-	-	-	-	**7.333**	**0.4159**