The STAP System for CN-Celeb Speaker Recognition Challange 2022

Shiwei Jiang, Junyu Chen, Qian Liu, Yuhua Qian *

Institute of Big Data Science and Industry, Shanxi University, China

jiangswtsf@163.com, jade.chenjunyu@gmail.com, stitch0507@163.com, jinchengqyh@126.com

Abstract

This report describes our submission to the speaker verification (fixed track and open track) of the CN-Celeb Speaker Recognition Challenge 2022 (CNSRC 2022). Both tracks share the same speaker verification system, which only uses CN-Celeb.T as our training set. In this challenge, our sub-systems consist of three styles of models, ResNet-based, TDNN-based, and Transformer-based. We also applied a series of approaches including data augmentation, attention module, loss function, and attention-based back-end for multiple enrollment utterances to enhance the corresponding sub-systems. Our final system is a fusion of 5 sub-systems and performed well on both tracks of CNSRC 2022. The minDCF of our submission is 0.3399, and the corresponding EER is 5.728%. In addition, observations on the gain of each method and the performance of one system under different genres were given.

1. Data

1.1. Datasets

The development of CN-Celeb1 dataset[1] contains 797 speakers with a total of 107,953 utterances and the CN-Celeb2 dataset[2] contains 1996 speakers with a total of 524,787 utterances. [2] showed there are a large amount short-time utterances that are considered hard samples in both datasets. After auditioning these samples, we found the size of the shortest ones is 26K, with a duration of less than 1 second, which means they are weakly discriminative or even noise samples. Therefore, we argued that these samples are not conducive to model training and filtered them (2459 utterances in total). Finally, 2793 speakers and 630,281 utterances were used for model training.

2. Models

2.1. Data Augmentation and Features Extraction

2.1.1. Data Augmentation

Data augmentation contains offline mode and online mode.

For the offline mode, we used a 3-fold speed augmentation [3] to generate extra twice speakers. Each utterance in this dataset was perturbed by 0.9 or 1.1 factor based on the SoX speed function. As a result, we obtained 8,379 speakers and 1,890,843 training speech utterances.

For the online mode, we adopted a strategy similar to that in [3] to construct an augmentation chain as:

- gain augment with a probability of 0.2
- white noise augment with a probability of 0.2
- noise addition augment with a probability of 0.6
- time stretch augment with a probability of 0.2

* Corresponding author.

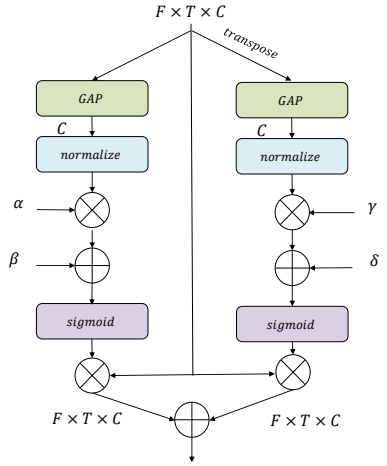

Figure 1: GCFA: gated channel-frequency attention module

Notably, we sampled some speech trials randomly by genre type from the CN-Celeb2 dataset and used them instead of the MUSAN dataset [4] to mix with the training utterances as noise addition augmentation. Three to seven recordings are randomly picked from genre speech, then added to the original signal from 13 to 20dB SNR. We also adopted SpecAugment[5] on log Mel spectrogram with randomly masking 0 to 10 frequency channels and 0 to 5 time frames.

2.1.2. Features Extraction

We extracted 80-dimensional log Mel spectrogram based on touchaudio. The window size is 25 ms, and the frameshift is 10 ms. 400 frames of features were extracted for transformer-based and TDNN-based sub-systems, and 250 frames of features were extracted for ResNet-based sub-systems. No voice activation detection (VAD) or speech enhancement was used.

2.2. Front-end Backbone

TDNN-based and ResNet-based models are the mainstream backbones used in speaker recognition with a good trade-off between computation and performance. Recently, transformer-based models have emerged in this field and achieved state-of-the-art performance [6]. Therefore, we adopted all three architectures simultaneously to construct various sub-systems.

2.2.1. ResNet-based

We constructed a new ResNet model with 40 layers following the design of ResNet34[7] and modified it by referring to [8, 9]. Then, we proposed a gated channel-frequency attention

module(GCFA) to enhance this model. The core design of our ResNet is as follows:

1. The number of blocks and channels in each stage is (2,2,12,2) and (64,96,192,384) respectively.
2. Remove the activation function in stem and at the end of each residual block.
3. Use the convolution with kernel size of 2 and step size of 2 to downsample the feature maps between stages.

Inspired by GCT [10], we proposed GCFA, a new gated frequency-channel context modeling attention module (as shown in Figure 1) to get better frequency attention and channel attention in ResNet. We added it before the residual connection in each residual block. The calculation steps of it are:

- **Gated Frequency** Firstly, feature maps $X \in \mathbb{R}^{F,T,C}$(C, F, and T represent the dimension of channels, frequency, and time) is pooled into a vector $\boldsymbol{s} \in \mathbb{R}^{F}$ containing frequency information by global average pooling (GAP). Next, scaled normalization is applied on $\boldsymbol{s}$ to get $\hat{\boldsymbol{s}} \in \mathbb{R}^{F}$.

$$\hat{s}_i = \frac{ks_i}{\|\boldsymbol{s}\|_2} = \frac{ks_i}{\sqrt{\sum_1^F {s_i}^2 + \epsilon}}, i \in \{i, ..., F\} \quad (1)$$

  where ϵ is 1e-6. k is a soft scale and is set to $\sqrt{F}$ in experiments.

  Then, the trainable gated vector $\boldsymbol{\alpha} \in \mathbb{R}^{F}$ and bias vector $\boldsymbol{\beta} \in \mathbb{R}^{F}$ are calculated with $\hat{\boldsymbol{s}}$ to get gated frequency vector $\boldsymbol{f} \in \mathbb{R}^{F}$ as follows:

$$f_i = \alpha_i \hat{s}_i + \beta_i, i \in \{1, ..., F\} \quad (2)$$

- **Gated Channel** Calculating process of gated channel vector is similar to that of gated frequency vector except the receptive field of the attention module is limited to the channel dimension. The process is as follows:

$$\hat{g}_i = \frac{kg_i}{\|\boldsymbol{g}\|_2} = \frac{kg_i}{\sqrt{\sum_1^F {g_i}^2 + \epsilon}}, i \in \{i, ..., C\} \quad (3)$$

$$c_i = \gamma_i \hat{g}_i + \omega_i, i \in \{1, ..., C\} \quad (4)$$

  where $\boldsymbol{g} \in \mathbb{R}^{C}$ is result of GAP along channel axis. $\gamma \in \mathbb{R}^{C}$, $\omega \in \mathbb{R}^{C}$ respectively represent the gating vector and bias vector in the channel dimension, and $\boldsymbol{c} \in \mathbb{R}^{C}$ represents the channel vector after gating.

- **Gated Frequency-Channel Attention** The gated frequency vector and the gated channel vector are combined with $sigmoid$ activation function and broadcasting to get gated frequency-channel attention which the original feature maps $X^{F,T,C}$ element-wise multiply.

$$\hat{X} = X(sigmoid(\boldsymbol{f}) + sigmoid(\boldsymbol{c})) \quad (5)$$

2.2.2. *TDNN-based*

Time Delay Neural Networks (TDNNs) hav become the mainstream approach in speaker verification tasks. TDNNs depend on 1D convolutions to capture global frequency feature, yet need many filters to model the fine details of any frequency region [11]. To mitigate this issue, many researchers introduce 2D convolutions before TDNN layers to incorporate frequency translational invariance. We applied two kinds of state of the art TDNNs based on 2D convolutions:

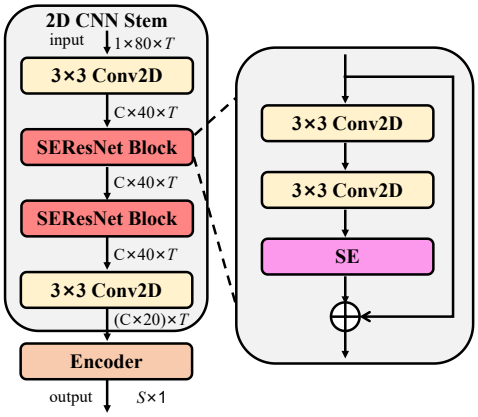

Figure 2: Structure of the CNN-Encoder

- **ECAPA CNN-TDNN** [11] This work added 2D convolutional stem to model high resolution frequency details in front of TDNN layers. Our ECAPA CNN-TDNN is similar to the vanilla, but we applied extra SE module in 2D convolution stem. Its structure is shown in Figure 2, where encoder is ECAPA-TDNN.
- **MFA-ECAPA** [12] To address the performance of TDNNs may degrade under short utterance scenaris, Liu et al. proposed MFA-ECAPA model with a novel dual-path design. We reproduced MFA-ECAPA as our another backbone with the kernel size of 3 and stride of 1 in TDNN layer of MFA module.

For all TDNN-based models, the number of channels C in the convolutional stem is 128, and the number of channels in the ECAPA-TDNN is 1024. The stride of the first convolutional layer of ECAPA is 3. The embedding dimension is 192.

2.2.3. *Transformer-based*

Currently, some researchers have found that combining global information is beneficial for convolutional models to get better performance [13, 14]. However, these works either decoupled the process of global interaction from local interaction[14], or only utilized simple contextual information[13, 15, 16]. Thus, we proposed a new architecture, Speaker-ViT, which combines the global modeling capability of the transformer architecture with the local modeling capability of the convolution to fully fuse the local and global modeling processes within and between tokens by stacking multiple global-local blocks. Its architecture is shown in the Figure 3. In the Pre-Norm structure, we chose batch normalization as Norm operation. LCN is referred from [8] for local modeling. MHSA is multi-head self-attention. TokenSE means calculate SE attention between channels of each token with shared weights, and DWConv is depthwise convolution. There is also a positional encoding mechanism to record the positional relationships between tokens and we add it to features after the first global-local block. Its calculation process is as follows:

$$y = \text{ELU}(\text{DWConv}(x)) \quad (6)$$

where ELU is exponential linear unit activation function DWConv is depthwise convolution. x is the input features and y is the positional encoding.

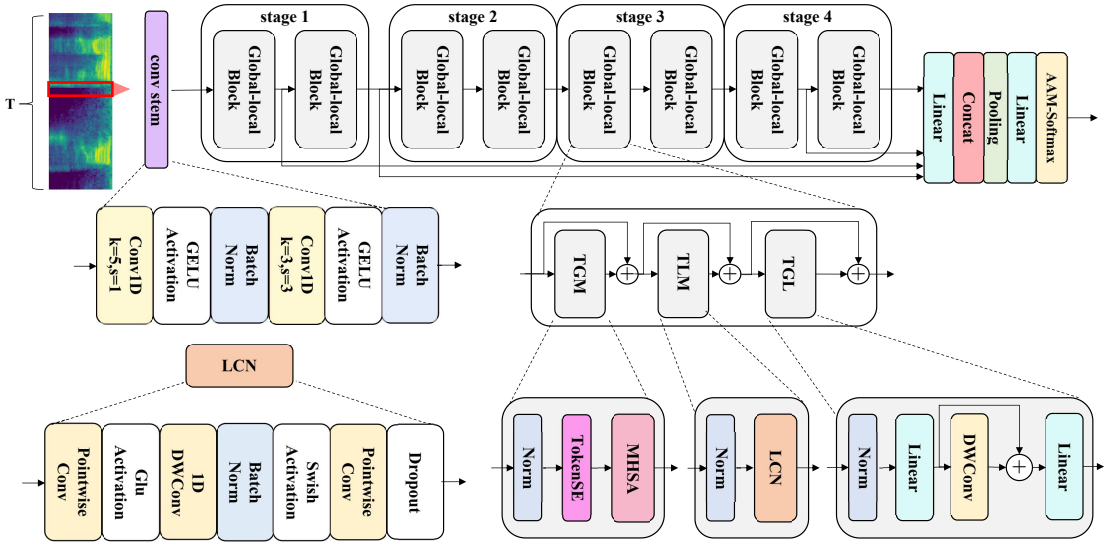

Figure 3: The overall architecture of Speaker-ViT

The low-level features containing details concatenate with the high-level semantic features to get comprehensive features after being reduced in dimension (divided by 2) through a weight-shared linear layer. The dimension of each token (channels) and embedding are both 400. The dimension scale in TGL is 2. The number of heads and dimension of each head are 8 and 64 respectively. Unless mentioned, the kernel size of all convolution is 3 and activation function is GELU in this model.

Inspired by [8], we also replaced the original stem in Speaker-ViT with 2D convolution layers to obtain more fine-grained local features in the spectrogram, which shows in Figure 2 where Encoder is Speaker-ViT without stem, C is 24, the kernel size of the first convolution is 5, and the stride of the last convolution is 3.

2.3. Pooling Method

The pooling layer aims to aggregate the variable frame-level features to a fixed utterance-level embedding. We applied channel-dependent attentive statistics (CAS) [13] as the pooling layer for all systems. This method can be described as below:

Suppose obtained frame-level feature $h = [h_1, h_2, ..., h_T]$, with $h_t \in \mathbb{R}^d$. We can gain a channel- and context-dependent scalar weight through a softmax layer:

$$\alpha_{t,c} = \frac{\exp\left(\boldsymbol{v}_c^T f(\boldsymbol{W}\boldsymbol{h}_t + \boldsymbol{b}) + k_c\right)}{\exp\left(\sum_\tau^T \boldsymbol{v}_c^T f(\boldsymbol{W}\boldsymbol{h}_\tau + \boldsymbol{b}) + k_c\right)} \tag{7}$$

where the parameters $\boldsymbol{W} \in \mathbb{R}^{R\times C}$, $\boldsymbol{b} \in \mathbb{R}^{R\times 1}$ and $\boldsymbol{v}_c \in \mathbb{R}^{R\times 1}$. The score $\alpha_{t,c}$ represents the importance of each frame given the channel c. Then, calculate the weighted mean and standard deviation of channel c and concatenate them to get pooled representation. For all systems, R is 128.

2.4. back-end

Since CN-Celeb.E contains multiple enrollment utterances, an attention-based back-end can better fuse multiple embeddings of enrollment than averaging them directly. So we adopted the attention-based back-end for multiple enrollment utterances mentioned in [17] to improve the front-end models. We set hyperparameters $d1 = 8$, $d2 = 8$, $D2 = 128$, and scale the dimension of input from D to $d1 * \mathrm{D}_{head}$ ($\mathrm{D}_{head} = 256$) during first fusion stage referring to the multi-head self attention mechanism.

2.5. Loss Function

2.5.1. AdaFace-Softmax

Magin-based loss functions have been widely chosen in speaker recognition, such as AM-Softmax [18] and AAM-Softmax [19]. Although they increase the inter-class margin and reduce the intra-class margin, they don't explicitly consider sample quality. It is well-known that low-quality unrecognizable samples have a nontrivial pernicious impact on the models with margin-based loss, and the CN-Celeb dataset includes more low-quality samples compared to the VoxCeleb dataset [20]. Therefore, we adopted the loss function proposed by [21], which considers the influence of sample quality on the margin. The formulation is given by (for more details, please refer to [21]):

$$\mathcal{L} = -\log \frac{\exp\left(f\left(\theta_{i,y_i}, m\right)\right)}{\exp\left(f\left(\theta_{i,y_i}, m\right)\right) + \sum_{j\neq y_i}^{n} \exp\left(s\cos\theta_{i,j}\right)} \tag{8}$$

$$f\left(\theta_{i,j}, m\right)_{\text{AdaFace}} = \begin{cases} s\cos\left(\theta_{i,j} + g_{\text{angle}}\right) - g_{\text{add}} & j = y_i \\ s\cos\theta_{i,j} & j \neq y_i \end{cases} \tag{9}$$

We also introduced subcenter method [22] to alleviate the effect of noisy and low-quality samples. The formulation is given by:

$$\cos\left(\theta_{i,j}\right) = \max_{1\leq k\leq K}\left(\|\boldsymbol{z}_i\| \cdot \|\boldsymbol{W}_{j,k}\|\right) \tag{10}$$

This loss function was used for the transformer-based and TDNN-based architecture to train the corresponding subsystems. s, m, h as the hyperparameters is 32, 0.2, 0.333 and the number of subcenters is 2 in experiments.

Table 1: Ablation Study on Speaker-ViT with 2793 speakers. + here denotes stacking our methods.

Methods	CN-Celeb.E(TTA)		CN-Celeb.E(Full)	
	EER(%)	minDCF	EER(%)	minDCF
Speaker-ViT	8.178	0.4629	8.077	0.4471
+ genres noise	7.823	0.4505	7.756	0.4401
++ Subcenter-AdaFace	9.012	0.4318	8.955	0.4193
+++ back-end	7.282	0.4240	7.215	0.4136
+++ CNN-Encoder	8.679	0.4228	8.522	0.4161
++++ back-end	7.102	0.4186	7.091	0.4112

2.5.2. Data Uncertainty Learning Loss

We utilized the data uncertainty learning loss (DUL loss) proposed by [23] to alleviate the negative impact of complex scenarios on speaker embeddings. Specifically, two linear heads are defined after the aggregation layer to output speaker embedding μ_i and uncertainty information δ_i(scene, noise, semantics). Reparameterization trick was used to keep gradients as usual. Specifically, we first sampled a random noise ϵ from a normal distribution, which is independent of the model parameters, and then generated $e_i$the equivalent sampling representation.

$$e_i = \mu_i + \epsilon\delta_i, \epsilon \in N(0, I) \tag{11}$$

To prevent δ_I degenerating to constant vector in the learning process, KL divergence was used as the regularization term to explicitly constrain δ_i. The definition of DUL loss function is shown as follows:

$$L_{DUL} = L_{class}(e_i) + \lambda KL(N(z_i|\mu_i, \delta_i^2)||N(\epsilon|0, I)) \tag{12}$$

λ is a hyperparameter and it is 1e-4 in experiments.

2.6. Training Protocol

We trained front-end embedding models and back-end models in turn.

In the front-end training stage, we used the SGD optimizer with a momentum of 0.9 and weight decay of 1e-4 for transformer-based models (2e-5 for TDNN-based models), and we used the RangerLars optimizer[24, 25] with a weight decay of 2e-5 for ResNet-based models. The initial learning rate is 4e-2 for transformer-based models, 3e-2 for TDNN-based models and 1e-2 for ResNet-based models. We adopted a cosine scheduler which steps at each iteration to decay the learning rate except the first epoch for warm-up. Batch size is 256 for 2793 speakers and 512 for 8379 speakers. We applied batch augmentation, similar to that of [26] and M is 4. The training stopped after 20 epochs for transformer-based models, 12 epochs for TDNN-based models, and 15 epochs for ResNet-based models.

In the back-end training stage, we only used 2793 speakers without noise addition augmentation to train our sub-systems. Differing from [17], we didn't freeze the weights of the front-end but finetuned them together with the back-end. A smaller finetuning learning rate of 8e-5 for the front-end and a larger learning rate of 2e-2 for the back-end was adopted. And we changed the frame size from 400(250) to 600 to fit long utterances. Batch size is 512 consisting of 4 random utterances sampled from each speaker (128 speakers in a batch). We only applied GE2E loss [27] for training. In addition, the learning rate scheduler is the same as in the first stage. The training process stopped after 12K iterations.

All the experiments were completed with Pytorch.

Table 2: Ablation Study on ResNet-Att with 2793 speakers. + here denotes stacking our methods.

Methods	CN-Celeb.E(TTA)		CN-Celeb.E(Full)	
	EER(%)	minDCF	EER(%)	minDCF
ResNet	7.789	0.4389	7.474	0.4300
+ GCFA	7.592	0.4238	7.535	0.4185
++ DUL loss	7.429	0.4206	7.311	0.4181

3. Results

3.1. ablation study

Firstly, we used the original Speaker-ViT backbone trained with 2793 speakers followed by AAM-Softmax (m = 0.25, s = 32) without multi-genres noise addition augmentation as the first baseline system. Then we studied the effectiveness of each method by stacking them gradually. Secondly, we also trained the improved ResNet model followed by AAM-Softmax (m = 0.25, s = 32) with the same datasets as the second baseline to study the effectiveness of GCFA module and DUL loss.

The performance was evaluated using the equal error rate (EER) and the minimum normalized decision cost function (minDCF) calculated where $C_{FA} = 1$, $C_{Miss} = 1$, and $P_{target} = 0.01$. Our systems included two test modes, extracting the whole utterances (denoted as Full) and TTA [20]. Once we tested the systems without back-ends, we directly averaged the multiple embeddings extracted from target speakers' enrollment to obtain their global embedding.

Table 1 shows the improvement of the Speaker-ViT baseline system by gradually stacking our proposed methods. First, the overall metrics are better after introducing multi-genres noise addition augmentation, especially under the TTA, with a 4.3% gain for EER and 2.7% gain for minDCF. Then, subcenter AdaFace-Softmax replacing AMM-Softmax also gets significant improvement. Although EER deteriorated remarkably on both test modes, minDCF achieved more than 4%. We argued that the subcenter enables the model to be more discriminant for positive samples, which may cause a slight increase in false reject rate (FR) and a significant decrease in false accept rate (FA). But EER is more sensitive to the change of FR due to the imbalance between positive and negative trials in the test set, thus causing the illusion of weak performance. Based on the settings above, we added the attention back-end after the front-end encoder and fine-tuned them together to get a superior system with EER increasing by around 20%. In addition, we combined the CNN-Encoder architecture mentioned in 2.2.3 with Speaker-ViT with channels of 480, and it is clear from the results that this architectural transformation is beneficial for generalization. However, we found slight overfitting in training, which indicated it could perform better with larger datasets.

Table 3: Performance of Sub-Systems and Fusion System.

Sub-Systems	**CN-Celeb.E(TTA)**		**CN-Celeb.E(Full)**	
	EER(%)	**minDCF**	**EER(%)**	**minDCF**
Speaker-ViT	9.012	0.4318	8.955	0.4193
S1: Speaker-ViT + back-end	7.282	0.4240	7.215	0.4136
ResNet-Att	7.502	0.4234	7.232	0.4174
S2: ResNet-Att + back-end	7.187	0.4039	7.170	0.4023
CNN-ECAPA-TDNN	8.037	0.4137	7.829	0.4076
S3: CNN-ECAPA-TDNN + back-end	6.928	0.4048	7.006	0.4007
MFA-ECAPA	8.237	0.4162	8.127	0.4062
S4: MFA-ECAPA + back-end	**6.860**	0.3974	**6.905**	0.3932
CNN-Speaker-ViT	7.851	0.3957	7.885	0.3922
S5: CNN-Speaker-ViT + back-end	6.888	**0.3832**	6.911	**0.3771**
Fusion System	**EER(%)**		**minDCF**	
S1 $\sim$ S5 (submited)	**5.728**		**0.3399**	

Table 4: Performance under Cross-Genre and Same-Genre Conditions.

	Advertisement	Drama	Entertainment	Interview	Live Broadcast	Movie	play	Recitation	Singing	Speech	Vlog
Cross	0.6902	0.9065	0.5339	0.7520	0.5745	0.9286	-	-	0.9055	0.4797	0.9560
Same	0.3333	0.2714	0.3815	0.2707	0.2223	0.5690	0.0800	-	-	0.1167	0.3523
Total	0.6762	0.3354	0.4112	0.3092	0.2389	0.5553	0.5336	0.1811	0.9055	0.1374	0.3428

Table 2 shows the improvement of the ResNet baseline system by gradually stacking a series of approaches. First, most of metrics have improved with GCFA module, especially under the TTA, with a 2.55% gain for EER and 3.44% gian for minDCF. Last, by replacing AAM-Softmax loss with DUL loss, the performance of all metrics has been improved.

3.2. Sub-Systems and Fusion Performance

All our sub-systems trained with 8379 speakers except S1 and fusion performance were described in Table 3. We found that applying a CNN stem before capturing global frequency feature module, such as Speaker-ViT, and ECAPA-TDNN seemed to obtain a better result. At the same time, we also found that most of systems with full-length utterance inputs can gain a better result compared to systems under TTA. Nearly half of the test utterances of test trials are less than four seconds long, which may be the cause of TTA's poor performance. For all sub-systems, introducing back-end brings performance gains, especially for EER. As our final submission for fixed and open track, we applied the fusion of these systems (S1$\sim$S5). The results of the fusion are shown in the last row of Table 3. The final metrics of our best fusion on the evaluation data are **5.728%** EER and **0.3399** minDCF.

3.3. Cross-genres Performance

We split the whole test trials into 22 sub-trials according to the genre of utterances and whether the genre is included or not in the enrollment of the target speakers, and then we utilized the CNN-Speaker-ViT sub-system to extract the embeddings of full-length utterances to calculate the minDCF for each sub-trials. The results showed in Table 4 where "cross" means the genre of speech is out of enrollment and "same" is vice versa. We dropped some sub-trials that were missing or invalid for calculating minDCF. From the results, minDCF on the cross-genre sub-set is consistently higher than that on the same-genre sub-set though the difference in each speaker's enrollment may lead to some bias in the results. We used samples from every genre to train our model and even sampled different genres for noise addition augmentation to capture intrinsic features. However, characteristics of genres still dominate in the extracted embeddings and lead to a seriously negative impact on the recognition in cross-genre scenarios. In terms of genre, the genres such as singing, movie, advertising, and playing possess discriminant characteristics and differ obviously from the other genres, so improving the performance across these genres is a significant challenge.

4. Conclusion

In this challenge, first, we modified the ResNet architecture and proposed a new attention module, GCFA, to build a new ResNet model. Second, we proposed a new backbone, Speaker-ViT, based on the transformer to fuse locality and globality. We also introduced AdaFace, DUL loss, and attention-based back-end to improve the robustness of the systems. Then, we ensembled several sub-systems based on three different architectures, ResNet, TDNN, and transformer, to obtain good results on speaker verification (fixed track and open track) of CNSRC 2022. The final result of our system was 0.3399 minDCF and 5.728% EER. In addition, observations on the gain of each method and the performance of one system under different genres were given.

5. References

[1] 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.

[2] Lantian Li, Ruiqi Liu, 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.

[3] Zhao Miao, Ma Yufeng, Liu Min, and Xu Minqiang, "The speakin system for voxceleb speaker recognition challange 2021," *ArXiv*, vol. abs/2109.01989, 2021.

[4] David Snyder, Guoguo Chen, and Daniel Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[5] Daniel S. Park, 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," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[6] Yang Zhang, Zhiqiang Lv, Haibin Wu, Shanshan Zhang, Pengfei Hu, Zhiyong Wu, Hung yi Lee, and Helen M. Meng, "Mfa-conformer: Multi-scale feature aggregation conformer for automatic speaker verification," *ArXiv*, vol. abs/2203.15249, 2022.

[7] Hee Soo Heo, Bong-Jin Lee, Jaesung Huh, and Joon Son Chung, "Clova baseline system for the voxceleb speaker recognition challenge 2020," *arXiv preprint arXiv:2009.14153*, 2020.

[8] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.

[9] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie, "A convnet for the 2020s," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[10] Zongxin Yang, Linchao Zhu, Yu Wu, and Yi Yang, "Gated channel transformation for visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11794–11803.

[11] Jenthe Thienpondt, Brecht Desplanques, and Kris Demuynck, "Integrating Frequency Translational Invariance in TDNNs and Frequency Positional Information in 2D ResNets to Enhance Speaker Verification," in *Proc. Interspeech 2021*, 2021, pp. 2302–2306.

[12] Tianchi Liu, Rohan Kumar Das, Kong Aik Lee, and Haizhou Li, "Mfa: Tdnn with multi-scale frequency-channel attention for text-independent speaker verification with short utterances," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7517–7521.

[13] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 3830–3834.

[14] Xiaoxiao Miao, Ian McLoughlin, Wenchao Wang, and Pengyuan Zhang, "D-mona: A dilated mixed-order non-local attention network for speaker and language recognition," *Neural Networks*, vol. 139, pp. 201–211, 2021.

[15] Yu-Jia Zhang, Yih-Wen Wang, Chia-Ping Chen, Chung-Li Lu, and Bo-Cheng Chan, "Improving Time Delay Neural Network Based Speaker Recognition with Convolutional Block and Feature Aggregation Methods," in *Proc. Interspeech 2021*, 2021, pp. 76–80.

[16] Sarthak Yadav and Atul Rai, "Frequency and temporal convolutional attention for text-independent speaker recognition," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6794–6798.

[17] Chang Zeng, Xin Wang, Erica Cooper, Xiaoxiao Miao, and Junichi Yamagishi, "Attention back-end for automatic speaker verification with multiple enrollment utterances," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6717–6721.

[18] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[19] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4685–4694.

[20] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, pp. 101027, 2020.

[21] Minchul Kim, Anil K Jain, and Xiaoming Liu, "Adaface: Quality adaptive margin for face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

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

[23] Jie Chang, Zhonghao Lan, Changmao Cheng, and Yichen Wei, "Data uncertainty learning in face recognition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5709–5718.

[24] Less Wright, "Ranger - a synergistic optimizer.," *GitHub repository*, 2019.

[25] You Yang, Gitman Igor, and Ginsburg Boris, "Scaling SGD batch size to 32k for imagenet training," *CoRR*, vol. abs/1708.03888, 2017.

[26] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry, "Augment your batch: Improving generalization through instance repetition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8126–8135.

[27] Li Wan, 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)*, 2018, pp. 4879–4883.