



AntVoice Neural Speaker Embedding System for FFSVC 2020

Zhiming Wang, Furong Xu, Kaisheng Yao, Yuan Cheng, Tao Xiong, Huijia Zhu

Ant Group, Shanghai, China

{zhiming.wzm, booyoungxu.xfr, kaisheng.yao, chengyuan.c, weilue.xt, huijia.zhj}@antgroup.com

Abstract

This paper presents a comprehensive description of the AntVoice system for the first two tracks of far-field speaker verification from single microphone array in FFSVC 2020 [1]. The system is based on neural speaker embeddings from deep neural network-based encoder networks. These encoder networks for acoustic modeling include 2D convolutional residual-like networks that are shown to be effective on the tasks. Specifically, we apply the Squeeze-and-Excitation residual network (SE-ResNet) [2] to model cross-channel inter-dependency information. On short utterances, we observe that SE-ResNet outperforms alternative methods in the text-dependent verification task. The system adopts a joint loss function that combines the additive cosine margin softmax loss [3] with the equidistant triplet-based loss[4]. This loss function results in performance gains with more discriminative speaker embeddings from enhanced intra-class similarity and increased inter-class variances. We also apply speech enhancement and data augmentation to improve data quality and diversity. Even without using model ensembles, the proposed system significantly outperforms the baselines [1] in both tracks of the speaker verification challenge. With fusion of several encoder neural networks, this system is able to achieve further performance improvements consistently. In the end, the AntVoice system achieves the third place in the text-dependent verification task.

Index Terms: far-field speaker verification, FFSVC, Squeeze-and-Excitation network, additive cosine margin softmax loss, equidistant triplet-based loss

1. Introduction

In this paper, we describe the AntVoice system, developed by Ant Group, for the 2020 Far-Field Speaker Verification Competition (FFSVC 2020) [1]. The competition aims to address challenges related to far-field speaker verification tasks. It is separated into tasks for 1) far-field text-dependent speaker verification from single microphone array, 2) far-field text-independent speaker verification from single microphone array, and 3) far-field text-dependent speaker verification from distributed microphone arrays. To simulate real usage scenarios, various background noise is played during recording, and enrollment utterances are collected in close-talking microphones while testing utterances in far-field microphone arrays. As a consequence, participants need to develop systems that are robust to background noise, room reverberations, mismatch between recording channels, *et al.* Our team has participated in tasks 1 and 2, which both involve speaker verification from single microphone array.

The field of speaker verification has advanced a lot due to the development of speaker embeddings using deep neural network [5, 6]. This paradigm has been very effective, achieving state-of-the-art results on speaker verification benchmark

data sets. It generally consists of a pooling layer, on top of encoded frame-level feature representations, to obtain segment-level information of speaker speech. Various temporal pooling techniques have been investigated [5, 7, 8, 9]. Once the segment-level information is obtained, it is mapped via feed forward network classifier to corresponding speaker ids. Although the effectiveness of this paradigm was demonstrated for speaker verification in close-talking environments [10], we find in this paper that more advanced techniques need to be developed for far-field speaker verification tasks.

In this paper, we report our approach as follows. Section 2 describes the front-end and data augmentation that increases diversity of the sample data. In Section 3, we describe the neural speaker embeddings used in this competition. Results are reported in Section 4. Finally, in Section 5 we conclude the paper.

2. The Front-end and Data Augmentation

2.1. Input Feature

Audios are resampled to 16,000 Hz. 80-dimensional logarithm Mel filter banks are generated within a 25ms sliding window using a hop step size of 10ms; cepstral mean normalization(CMN) is performed within a 3-second sliding window (it would be degenerated into a fixed one that covers all frames for shorter utterances such as those less than 3 seconds), and then followed with cepstral mean and variance normalization(CMVN) within the whole utterance. Energy based voice activity detector(VAD), similar to the one in Kaldi [11], is employed to remove silence frames; here we use adaptive threshold per utterance, that is, 1.0325 times the average energy of the head-and-tail respective 30 frames. Other acoustic feature configurations are the same as the default setups in Kaldi [11].

During training, chunks of 1 ~ 4.5 second long audio segments are randomly sampled from recordings to have a constant L number of frames in a mini-batch. In the first task, the frame length L is uniformly sampled within the interval of [100, 256] and the interval is [200, 450] in the second task for longer utterances. These frames then form a batch of acoustic features of size $B \times L \times 80$, where B is the mini-batch size and set to 32 in the experiments.

2.2. Speech Enhancement and Data Augmentation

To enhance speech quality, we use the weighted prediction error(WPE)[12, 13] algorithm to reduce signal dereverberation for enrollment and testing recordings as [14].

To reduce mismatching between close-talking and far-field speech and improve model robustness, we augment the training recordings that are originally from the microphone arrays or the cellphone provided by the challenge organizer. To be specific, we use image source model(ISM) based Pyroomacoustics [15] to simulate the room impulse response(RIR) generator. With this method, each utterance generates five replicas

Table 1: The convolutional ResNet-34 architecture.

Layers	Configurations
Conv1	$(3 \times 3, 64)$, stride (1×2)
Res1	$[(3 \times 3, 64)_2] \times 3$
Res2	$[(3 \times 3, 128)_2] \times 4$
Res3	$[(3 \times 3, 256)_2] \times 6$
Res4	$[(3 \times 3, 512)_2] \times 3$
Conv2	$(3 \times 3, 512)$, stride (1×2)
Pooling	statistical pooling(mean + std dev)
Linear1	2048×512
Linear2	512×512 (embedding features)
Classifier	$512 \times \text{\#Spks}$

from diverse microphone arrays that are placed at distances of -1.5m, 1m, 3m, 5m, and left or right 3m far from the sound source as like recording configurations of FFSVC 2020 training audio samples [1]. Additionally, with the methods in Kaldi¹ [11], reverberation, background noise, music and babble are respectively mixed into the original training samples at random signal-to-noise ratio (SNR) between 0 to 20 dB, resulting in about 750,000 augmented audio samples.

Following [1, 14], we use the background noise of the testing utterances to perform enrollment augmentation trial by trial. Particularly, we adopt the above energy-based VAD method to detect non-speech regions of the 4 testing utterances; then these non-speech parts are mixed, as noise, with the corresponding enrollment utterance with the SNR of testing recordings. This produces one simulated enrollment utterance for every trial.

3. Neural Speaker Embeddings

3.1. Encoder Networks

Since 2017, methods to represent speaker characteristics are dominated by deep neural networks [5, 6]. These approaches generally use an encoder network to extract frame-level representations from acoustic features, e.g., Mel filter banks or MFCC. The frame-level representations are followed with a pooling layer to aggregate them into segment-level speech characteristics. Finally, a fully connected classification network projects the extracted segment-level representations to corresponding speaker ids. We term the segment-level speaker characteristics as neural speaker embeddings or x-vectors. It is important to encode enough discriminative information in the embeddings to distinguish different speakers. We introduce the encoder networks in our experiments as follows.

E-TDNN and F-TDNN. We use E-TDNN and F-TDNN as in [16], but with the following differences: Exponential Linear Unit(ELU) which is defined as $f(x) = \max(0, x) + \min(0, e^x - 1)$, rather than ReLU, is used as the nonlinear activation; for the outputs of each parameter layer, batch normalisation is applied before the nonlinearity; the nonlinear activations of size 512 at the penultimate layer are used as embedding features for verification tasks as in [10]. These distinctions are also applied in ResNet and SE-ResNet below.

ResNet. We use standard 34-layer convolutional residual network (ResNet) architecture [17] in our experiments, as is described in Table 1. In Table 1, $[(3 \times 3, 64)_2] \times 3$ means 3 residual blocks, one of them consisting of 2 convolutional layers with kernel size of 3×3 and 64 filters; other setups are in

Table 2: The “Squeeze-and-Excitation” block of SE-ResNet.

Layers	Configurations
Linear1	$N_c \times \max(N_c/16, 32)$
Nonlinear1	ELU
Linear2	$\max(N_c/16, 32) \times N_c$
Nonlinear2	Sigmoid

analogy. For the first block of Res2 \sim 4 with different numbers of channels between the input and output, a short cut connection between them is needed via a convolutional layer with kernel size of 1×1 .

SE-ResNet. To model cross-channel interactions in convolutional network, we use “Squeeze-and-Excitation” residual network(SE-ResNet) [2]. In SE-ResNet, the SE block adaptively re-calibrates per channel feature responses by explicitly modelling inter-channel dependency, corresponding to channel-wise attention. The convolutional residual parts of SE-ResNet are the same as in Table 1, and the SE block of SE-ResNet is described in Table 2 with N_c denoting the number of channels.

3.2. Loss Function

We use a joint loss function that is the summation of additive cosine margin softmax loss and equidistant triplet-based loss.

Additive Cosine Margin Softmax Loss. Additive cosine margin softmax loss, aka *CosAMS*, was proposed in [3]. With embedding feature x_j from the j -th sample of a mini-batch and the constraint of zero bias in the classifier layer, $\cos(\theta_{\langle x_j, w_c \rangle}) = \frac{w_c^T x_j}{\|w_c\| \cdot \|x_j\|}$, in which w_c is the weight vector corresponding to speaker class c , $\theta_{\langle x_j, w_c \rangle}$ is the angle between x_j and w_c , and $\|\cdot\|$ is the l_2 -norm. The *CosAMS* loss is defined as follows

$$L_{CosAMS} = -\frac{1}{B} \sum_{j=1}^B \log \frac{e^{\eta(\cos(\theta_{\langle x_j, w_{y_j} \rangle}) - m)}}{Z_{x_j}},$$

$$Z_{x_j} = e^{\eta(\cos(\theta_{\langle x_j, w_{y_j} \rangle}) - m)} + \sum_{i \neq y_j} e^{\eta \cos(\theta_{\langle x_j, w_i \rangle})},$$
(1)

where y_j is the label corresponding to x_j , η is a scale hyperparameter, and m is the margin. Using a larger η makes the posterior sharper than using $\eta = 1$; increasing m would result in reduced posterior in Eq. (1), therefore forcing x_j to be more discriminative.

To avoid local optimum or divergence when training models with the discriminative loss functions as L_{CosAMS} , we use an annealing strategy on m to make training process stable [10]. Empirically, we increase the margin m linearly from 0 to the target margin value as $m = \min(m_{max}, m_{inc} \times \bar{e})$, where $\bar{e} \in [0, 1, 2, \dots]$ is the training epoch index. In our experiments, we set $\eta = 30$, $m_{inc} = 0.07$, $m_{max} = 0.25$.

Equidistant Triplet-based Loss. While traditional triplet loss aims to force the distance between the matched positive sample and the anchor less than that between the mismatched pairs by at least a presupposed margin α , equidistant triplet-based(or *EDTri* for short, [4]) loss further introduces equidistant constraint terms, which pull the matched samples closer by adaptively constraining two samples of the same class to be equally distant from another one of a different class in each triplet. By optimizing *EDTri* loss, the algorithm progressively maximizes intra-class similarity, contributing to generate more dis-

¹github.com/kaldi-asr/kaldi/blob/master/egs/sre16/v2.

Table 3: The hyper-parameters of cyclical learning rate.

Hyper-parameters	Pretraining	Finetuning
max_lr	10^{-3}	2.5×10^{-5}
base_lr	2.5×10^{-4}	6.25×10^{-6}
step_size_up	2 epochs	half an epoch

criminative embeddings.

To be specific, in a mini-batch, for a certain anchor sample x_a , we choose the closest mismatched sample x_n and the farthest matched sample x_p in the embedding feature space to form a triplet $\{x_a, x_p, x_n\}$, where the labels satisfy $y_p = y_a$ and $y_n \neq y_a$. The *EDTri* loss function is as

$$\begin{aligned}
L_{EDTri} &= L_{Tri} + L_{EquiD}, \\
L_{Tri} &= \frac{1}{B} \sum_{j=1}^B [d(x_{j,a}, x_{j,p}) - d(x_{j,a}, x_{j,n}) + \alpha]^+, \\
L_{EquiD} &= \frac{1}{B} \sum_{j=1}^B ([d(x_{j,a}, x_{j,p}) - d(x_{j,p}, x_{j,n}) + \alpha]^+ + \\
&\quad |d(x_{j,p}, x_{j,n}) - d(x_{j,a}, x_{j,n})|), \tag{2}
\end{aligned}$$

where d is the $l2$ -norm distance, $[\cdot]^+ = \max(\cdot, 0)$, $|\cdot| = \text{abs}(\cdot)$. In our experiments, we let $\alpha = 0.3$.

In conclusion, while *CosAMS* loss focuses on between-class separability by enlarging inter-class variances with margin-based methods, *EDTri* loss emphasizes more on explicitly reducing intra-class variability; each acts as a regularizer to the other.

3.3. Model Training

Inspired by the transfer learning strategy in [14] and following the FFSVC 2020 Challenge’s data protocol [1], we use the corpora in OpenSLR² (including SLR18, SLR33, SLR38, SLR47, SLR49, SLR62 and SLR68, in total of 9127 speakers) to pre-train deep encoder models in Sec. 3.1 for speaker embeddings. Then the models are finetuned with domain-dependent data sets as in [1]: in the first task, we use the HI-MIA data set (SLR85, [18]) and the first 30 utterances of FFSVC 2020 training data set; in the second task, we use the remaining FFSVC 2020 training data set. They are used together with their corresponding augmented audio utterances according to the procedure described in Sec. 2.

Models are trained using the RADAM optimizer [19], with the weight decay of 5×10^{-4} . The learning rate is scheduled with the cyclical strategy [20] that has two benefits: one to allow rapid traversal of saddle point plateaus and second to meet the optimum learning rate. The hyper-parameters³ of cyclical learning rate (CyclicLR) are listed in Table 3. Training samples are randomly shuffled at the beginning of each epoch. These training strategies are applied to all encoder networks as mentioned above.

²<http://openslr.org>.

³Their definitions are referred to PyTorch documentation on CyclicLR, pytorch.org/docs/stable/index.html.

Table 4: The weights of encoder networks for score fusion.

Task	E-TDNN	F-TDNN	ResNet	SE-ResNet
#1	0.13	0.15	0.26	0.46
#2	0	0.15	0.57	0.28

4. Experimental Results

4.1. Scoring Methods and Encoder Ensembles

We have experimented both cosine similarity and probabilistic linear discriminant analysis (PLDA) [21] as back-end scoring methods. Our preliminary experiments indicated that cosine similarity is superior to PLDA.

Utterances with the whole length are used for evaluation. In each trial, two enrollment utterances, including the simulated audio recording that is described in Sec. 2, are firstly whitened on their extracted embedding features, followed by length normalization, and finally averaged into the speaker embedding; the same procedure is also applied to the testing audios from 4 channels of the far-field microphone arrays.

As an ensemble strategy for submitting final results, the scores from different encoder networks are linearly weighted into a regression value, with the weights in Table 4. The weights are tuned on the development data set.

4.2. Main Results

The results for speaker verification on the development and evaluation data sets are reported in Table 5. Following the baseline system [1], we adopt minimum detection cost function (minDCF, with $P_{target} = 0.01$) as primary metric, and equal error rate (EER) as auxiliary one. Smaller values of them correspond to better system performances. We observe that, in both tasks, our methods perform significantly better than the baselines, even with a single model of ResNet or SE-ResNet. Furthermore, when using the fusion strategy and in comparison to the baselines, we could observe significant reductions of minDCF by 42.28% on the development set and 26.5% on the evaluation set in the first task; in the second task, the reductions were 13.97% and 16.94%. In the global leader-board (30% trials) of FFSVC 2020, the proposed system ranked the fourth in the first task, and ultimately officially ranked the third place with all the trials⁴; it ranked the fifth in the second task.

4.3. Analysis

4.3.1. Impact of encoder networks

From Table 5 and 4, we observe that residual-like networks such as ResNet and SE-ResNet roundly surpass TDNN-type ones by a large margin in performances⁵. That is because 2D convolution in both time and frequency axes is more effective than 1D time-axis convolution as in E-TDNN and F-TDNN.

Especially, we could see that the SE-ResNet model contributes much more, accounting for 46%, to the performance improvement in the first text-dependent verification task, which can be attributed to the cross-channel correlation modeling ca-

⁴According to the FFSVC 2020 evaluation plan [22], a fixed 30% of the evaluation trials were released in the leader-board, and the official final results were evaluated with all the trials.

⁵The comparison may not be fair for E-TDNN and F-TDNN of less number of weight parameters, but we use their best configurations as in [16] where increasing the number of parameters could not bring in more performance improvement.

Table 5: Performance results for speaker verification, with cosine similarity in general unless otherwise specified.

Models	Task#1		Task#2	
	minDCF	EER(%)	minDCF	EER(%)
Development data set				
baselines[1]	0.57	6.01	0.58	5.83
E-TDNN	0.456	4.32	0.685	6.76
F-TDNN	0.495	4.45	0.704	6.91
ResNet	0.427	3.58	0.557	4.9
SE-ResNet	0.394	3.12	0.569	5
SE-ResNet + PLDA	0.464	4.286	-	-
fusion	0.329	2.61	0.499	4.23
Evaluation data set(30% trials)				
baseline[1]	0.62	6.37	0.66	6.55
our submission	0.4557	4.25	0.5482	4.72

Table 6: The performance gains of the joint loss function.

Loss	minDCF	EER(%)
L_{CosAMS}	0.172	1.79
$L_{CosAMS} + L_{EDTri}$	0.163	1.41

pability that is critical to the performance on short utterances; but the same tendency could not be observed yet in the second text-independent task, which deserves more explorations.

4.3.2. Impact of the joint loss function

In addition to the main experiments, here we show that the joint loss function is more effective than using only $CosAMS$ loss. We used the development data set of Voxceleb2 for training, the original test set of VoxCeleb1 for testing (SLR49, [23]), *ResNet*-34 as the encoder network, and applied the pretraining strategy described in Sec. 3.3⁶. Results in Table 6 show that the proposed loss function, $L_{CosAMS} + L_{EDTri}$, leads to consistent performance improvements in comparison to using $CosAMS$ loss alone in terms of both minDCF and EER. The $EDTri$ loss is a metric learning method that does explicit reduction of intra-class variability. The results here verify that, a joint optimization of two discriminative loss functions that contributes to a more compact embedding space can dramatically reduces minDCF and EER.

The EER of 1.41% in Table 6 is obtained without model ensembles nor any data augmentation. As an additional advantage over large margin Gaussian mixture loss in [10], in which the number of parameters (mean and standard deviation per speaker) scales linearly with that of speakers, $EDTri$ loss does not incur extra memory overhead.

4.3.3. Impact of whitening

We observe that whitening, which is to subtract a mean vector from the embeddings, also plays a significant role in performance improvements. Particularly, we studied three schemes in the experiments of Sec. 4.3.2. Method **A** does not do whitening; method **B** acquires embedding mean vector for whitening from all whole-length training samples, and method **C** obtains such mean vector from test samples with whole lengths. Results in Table 7 show that whitening, especially with test-domain de-

pendent embedding mean, is effective at reducing EERs. The reason for that is whitening servers as a countermeasure to suppress redundant information in the embedding space. Method **B** is more appropriate in practice. Similar impacts from whitening are also observed in both tasks of FFSVC 2020, which are not shown here due to space limit.

Table 7: The impact of whitening.

Method	A	B	C
EER(%)	1.47	1.41	1.27

5. Conclusions

This paper describes the AntVoice system for far-field speaker verification tasks in FFSVC 2020. We use speech enhancement and data augmentation for improved data quality and diversity. We have observed that 2D convolutional residual-like networks in both time and frequency axes are more competitive than E-TDNN and F-TDNN based on 1D time-axis convolution. In particular, we observe that SE-ResNet for cross-channel modeling leads to consistent performance improvement in the text-dependent verification task targeting at short utterances. Moreover, we notice that a proposed combination of $CosAMS$ loss and $EDTri$ loss contributes to performance gains. This is attributed to their explicit increasing of intra-class similarity and inter-class variances that contribute to compact and discriminative speaker embeddings. With applications of these novel modeling methods and loss functions, the AntVoice system outperforms baselines, even with using a single model of ResNet or SE-ResNet. It wins the third place in the first text-dependent verification task of this challenge in the end.

6. References

- [1] X. Qin, M. Li, H. Bu, W. Rao, R. K. Das, S. Narayanan, and H. Li, "The interspeech 2020 far-field speaker verification challenge," *arXiv preprint arXiv:2005.08046*, 2020.
- [2] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.
- [3] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.
- [4] F. Xu, W. Zhang, Y. Cheng, and W. Chu, "Metric learning with equidistant and equidistributed triplet-based loss for product image search," in *The Web Conference (WWW)*, 2020, pp. 57–65.

⁶The training procedure is divided into two stages of pretraining and finetuning, not an end-to-end way, so we directly evaluate our proposed methods on VoxCeleb data set.

- [5] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2017, pp. 999–1003.
- [6] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.
- [7] F. R. R. Chowdhury, Q. Wang, I. LopezMoreno, and L. Wan, "Attention-based models for text-dependent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5359–5363.
- [8] G. Bhattacharya, J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification," in *Interspeech*, 2017, pp. 1517–1521.
- [9] Z. Wang, K. Yao, X. Li, and S. Fang, "Multi-resolution multi-head attention in deep speaker embedding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [10] Z. Wang, K. Yao, S. Fang, and X. Li, "Joint optimization of classification and clustering for deep speaker embedding," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 284–290.
- [11] D. Povey, A. Ghoshal, G. Boulianne, and et al., "The kaldi speech recognition toolkit," in *IEEE Automatic Speech Recognition and Understanding Workshop(ASRU)*, 2011.
- [12] T. Yoshioka and T. Nakatani, "Generalization of multi-channel linear prediction methods for blind mimo impulse response shortening," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 10, pp. 2707–2720, 2012.
- [13] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing," in *13. ITG 2018*, Oct. 2018.
- [14] X. Qin, D. Cai, and M. Li, "Far-field end-to-end text-dependent speaker verification based on mixed training data with transfer learning and enrollment data augmentation," in *Proc. Interspeech*, 2019, pp. 4045–4049.
- [15] R. Scheibler, E. Bezzam, and I. 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)*. IEEE, 2018, pp. 351–355.
- [16] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, L. P. García-Perera, F. Richardson, R. Dehak *et al.*, "State-of-the-art speaker recognition with neural network embeddings in nist sre18 and speakers in the wild evaluations," *Computer Speech & Language*, vol. 60, p. 101026, 2020.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [18] X. Qin, H. Bu, and M. Li, "Hi-mia: A far-field text-dependent speaker verification database and the baselines," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7609–7613.
- [19] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.
- [20] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 464–472.
- [21] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision(ECCV)*. Springer, 2006, pp. 531–542.
- [22] X. Qin, M. Li, H. Bu, R. K. Das, W. Rao, S. Narayanan, and H. Li, "The ffsvc 2020 evaluation plan," *arXiv preprint arXiv:2002.00387*, 2020.
- [23] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.