INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Investigation of IMU&Elevoc Submission for the Short-duration Speaker Verification Challenge 2021

Peng Zhang[1*], *Peng Hu*[2*], *Xueliang Zhang*[1]

[1]College of Computer Science, Inner Mongolia University, Hohhot, China
[2]Elevoc Technology Co., Ltd, Shenzhen, China

zhangpeng@mail.imu.edu.cn, peng.hu@elevoc.com, cszxl@imu.edu.cn

Abstract

In this paper, we present the IMU&Elevoc systems submitted to the Short-duration Verification Challenge (SdSVC) 2021. Our submissions focus on both text-dependent speaker verification (Task 1) and text-independent speaker verification (Task 2). First, we investigate several frame-level feature extractor architectures based on ResNet, Res2Net and TDNN. Then, we integrate Squeeze-Excitation block and dimension cardinality to further improve the Res2Net-based backbone network. In particular, we probe an effective transfer learning strategy that overcomes the lack of Task 1 datasets and improves in-domain performance. A knowledge distillation method fusing multiple models is proposed to obtain a stronger single model. Experimental results on the SdSVC 2021 show that our primary system yields 0.0500 MinDCF in Task 1 (ranked as 4th) and 0.0448 MinDCF in Task 2 (ranked as 6th).

Index Terms: speaker verification, transfer learning, knowledge distillation, SdSVC 2021

1. Introduction

The goal of speaker verification (SV) is to verify the speaker identity associated with the enrolled target speaker from the digital audio signal level [1]. Recently, SV has gained a lot of research interests with the advancement and popularity of virtual assistants such as Amazon Alexa, Apple Siri, Google Assistant, and Microsoft Cortana. Real applications have several requirements for the SV system. One thing is that SV should be robust to short duration speech segments, which is very important to user's experience. To improve the performance on short duration speech segments, several techniques have been proposed in previous studies [2, 3, 4, 5, 6, 7, 8, 9].

Currently, there are two main approaches for SV: i-vector [10] and deep embeddings. The development of deep neural networks (DNN) based speaker embedding has significant improvement. This mechanism has been developed through a variety of methods to generate discriminative speaker embedding, e.g. time-delay neural network (TDNN) [11, 12], convolutional neural network (CNN) [13, 14, 15, 16], and long short-term memory network [17].

In this paper, we describe the IMU&Elevoc team's submissions developed for the Short-duration Speaker Verification Challenge (SdSVC) 2021. The main goal of the SdSVC 2021 is to evaluate new technologies for text-dependent (TD) and text-independent (TI) speaker verification in short duration scenario. The challenge evaluates SdSVC with varying degree of phonetic overlap between the enrollment and test utterances (cross-lingual). The SdSVC 2021 includes two tasks. Task 1 is TD speaker verification, which constrains the transcripts of enrollment to a specific phrase. Task 2 is TI speaker verification, which only takes the speaker identities into account. Our work focus on both Task 1 and Task 2. The main contributions of this paper are summarized as follows:

1) We investigate several frame-level feature extractors using different neural network architectures to extract discriminative speaker embeddings, including ResNet, Res2Net and TDNN architectures. Afterwards, we propose the Squeeze-Excitation (SE) block and dimension cardinality to improve the performance of Res2Net-based architecture.
2) We probe different transfer learning strategies to deal with the problem of insufficient data in TD task. We found that fine-tuning the whole model achieves promising result, compared with fine-tuning some layers.
3) In order to obtain a single model with excellent performance, we employ knowledge distillation method to improve the performance of a single model with the fusion of multiple models and confirm its effectiveness.

The remainder of this paper is organized as follows: Section 2 introduces the methodology in our submissions. The experimental setup is presented in Section 3. Section 4 contains a discussion of the implication of the results of Section 2. We conclude this paper in Section 5.

2. Methodology

2.1. Speaker embedding

The speaker embedding can be decomposed into a frame-level feature extractor, a pooling layer and utterance-level feature extractor [12]. For our submissions, by fixing the attentive statistic pooling layer (ASP) [18] and utterance-level representation layers, we compare the frame-level feature extractor based several neural networks.

ResNet-based architecture. ResNet has been successfully applied to speaker recognition [16, 19] and is adopted here. We use ResNet-34 and ResNet-101 architectures for frame-level feature extraction. Different from the original ResNet, we integrate Squeeze-Excitation (SE) block [20] before the residual connections of the ResNet module. The SE block can adaptively re-calibrate channel-wise feature responses by explicitly modeling inter-dependencies among channels. In order to reduce computational cost, the channels in each residual block is half of compared to the original ResNet, i.e. the number of channels in the residual blocks used in our experiments are $\{32, 64, 128, 256\}$.

* equal contribution

Copyright © 2021 ISCA

http://dx.doi.org/10.21437/Interspeech.2021-743

Res2Net-based architecture. Different from ResNet, Res2Net [21] increases the range of receptive fields for bottleneck type residual block by introducing a more granular multi-scale structure. We mainly employ Res2Net-50 as our feature extractor backbone. Moreover, we integrate Res2Net with advanced modules to further improve the performances, including cardinality dimension [22], as well as SE block [20]. The dimension cardinality changes filters from single-branch to multi-branch and improves the representation ability of CNN. We replace the 3×3 convolution with the 3×3 group convolution, where the number of groups set 8 in our experiments. In addition, we reduce the channels of each residual block to $\{32, 64, 128, 128\}$, and set the scale dimension to 4.

TDNN-based architecture. Compared with the standard TDNN, ECAPA-TDNN [23] incorporates the SE blocks [20], 1-dimensional multi-scale Res2Net [21] layers, multi-layer feature aggregation [24] and channel-dependent attentive statistics pooling. Here, we choose the larger model (14.7 million parameters) in their paper for comparison, which has better performance.

2.2. Transfer embedding learning

Deep neural networks usually require large data for the generalization. For the TD task, a problem is lack of TD corpus which is expensive to collect.

In contrast, the dataset without text annotation is easier to collect and a large mount of TI data in the Task 1 can be used as training data. Therefore, we investigate the transfer learning strategy [25, 26] to adapt the TI model to the TD model instead of training the TD model from scratch.

As shown in Figure 1, we mainly investigate two effective transfer learning strategies in our submissions. The left one is to use the TI model as the pre-trained, and then fine-tune the entire model using TD data.

The right one is to directly copy the parameters of the frame-level feature extractor of the TD model to the TI model, freeze these parameters, and then fine-tune the pooling and the speaker embedding layer. It should be noticed that the parameters provided by the TI model do not include the speaker classification layer.

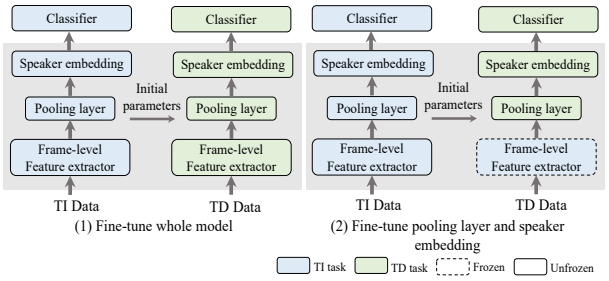

Figure 1: *Transfer the text-independent (TI) speaker model to text-dependent (TD) model.*

2.3. Knowledge distillation for fusion

Model fusion invariably is a powerful method that can improve performance on various learning tasks, which combines multiple independent models through strategies to enhance the effect of the single model. Score fusion and speaker embedding fusion are common methods for speaker recognition task. However, fusion takes too much costs, because it has to run multiple models simultaneously. To reduce the performance gap between to fusion multiple models and a single model, knowledge distillation is a natural approach [27].

We first train multiple networks independently, and take the average of all the speaker embeddings as the fusion results. Then, we train another single network using the fusion results of the multiple networks as the training target. This is referred as reacher-student learning.

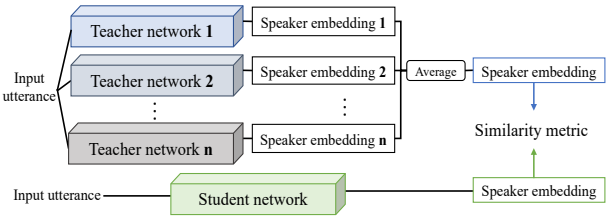

Figure 2: *A schematic of distilling the fused multiple models into a single model.*

It is intuitive to directly constrain the similarity between two embeddings learned from the teacher and student models [28]. Here, we employ minimum square error (MSE) or cosine distance (COS) loss as the optimization metric for speaker embedding distillation. Distillation loss functions are defined as:

$$L_{mse} = \frac{1}{M}\sum_{i=1}^{M}(\hat{e}_t^i - e_s^i)^2 \tag{1}$$

$$L_{cos} = -\frac{1}{M}\sum_{i=1}^{M}\frac{\hat{e}_t^i \cdot e_s^i}{\|\hat{e}_t^i\|\,\|e_s^i\|} \tag{2}$$

where $\hat{e}_t = \frac{1}{N}\sum_{j=1}^{N} e_t^j$,represents the average speaker embedding of N teacher models. e_s is the speaker embedding of student model. i represents i-th sample. The final loss function is $L_{spk} + \alpha L_{mse}$ or $L_{spk} + \beta L_{cos}$. L_{spk} is a criterion for speaker embedding learning, α and β are the corresponding weighting parameters. We set α and β to 0.5, according to our experiment. We also compare the performances of L_{mse} and L_{cos} in Section 4.3.

3. Experiments

3.1. Datasets

The SdSVC 2021 training and evaluation datasets originate from the DeepMine corpus [29]. The majority of the utterances are in Farsi (Persian) and a smaller subset in English. For both Task 1 and 2, the in-domain data provided by the challenge organizers is divided into training, development, enrollment and test partitions.

For Task 1, we only use the in-domain Task 1 training partition of the DeepMine dataset, which consists of 101,063 utterances from 963 different speakers. We use it to train phrase classification network and speaker embedding network.

For Task 2, we use three parts of the dataset:

- **DeepMine(Task 2 Train Partition)**. It consists of 124,901 utterances from 588 speakers. The utterances in training set are in Farsi and English;
- **VoxCeleb 1 & 2**. We use the development sets of VoxCeleb1 [30] and VoxCeleb2 [31], which consist of

148,642 and 1,092,009 utterances from 1,211 and 5,994 speakers respectively;

- **LibriSpeech**. We use the train-clean/other sets of LibriSpeech corpus [32], which comprise 281,241 utterances from 2,338 speakers.

The total number of utterances are 1,646,793 from 10,131 speakers. The official development datasets are used for tuning model parameters.

3.2. Front-End Processing

During training, we randomly cut 2-second temporal segments extracted from each utterance. Pre-emphasis with a coefficient of 0.97 is applied to the input signal. The spectrograms are extracted with a hamming window of 25 ms width and 10 ms frame shift.

The the input features of the TDNN are 80 dimensional Mel-frequency cepstral coefficients (MFCCs). For ResNet and Res2Net, 64 dimensional Mel-filterbanks are used. Mean and variance normalization is performed by applying instance normalization [33] to the input features.

In order to filter out the non-speech segments, voice activity detection (VAD) [34] is used in training and test stage. Since the dataset contains noise interference, we employ the complex spectral mapping speech enhancement [35] to remove the noise before VAD processing, which is trained by using the train-clean set of LibriSpeech, and MUSAN [36] corpus for noise data.

3.3. Online data augmentation

The SdSVC 2021 allows using other non-speech datasets for data augmentation purpose. We employ diverse additive noises and reverberations for data augmentation. The additive noises are selected from the MUSAN corpus; The reverberations are generated by using simulated small and medium room impulse responses [37]. Online data augmentation strategy is used to make the speaker embeddings more robust. The parameters selected for data augmentation are the same as those on the Kaldi recipe of [12].

3.4. Loss function

We use Circle Loss, proposed in [38], as the speaker loss function, which has outstanding performance in speaker verification task [39]. It is formulated as:

$$L_{spk} = -log\frac{exp(\gamma\alpha_p(s_p - \Delta_p))}{exp(\gamma\alpha_p(s_p - \Delta_p)) + \sum_{j=1}^{N-1} exp(\gamma\alpha_n^j(s_n^j - \Delta_n))}$$

where s_n stands for between-class similarity, as well as s_p indicates within-class similarity. N is the number of training speaker classes. γ is the scale factor, and m is the relaxation margin. $\alpha_p = [1 + m - s_p]_+$, $\alpha_n^j = [s_n^j + m]_+$, $\Delta_p = 1 - m$, $\Delta_n = m$. The Circle loss optimizes s_p and s_n separately with adaptive penalty strength and adds within-class and between-class margins. During training, we set $\gamma = 256$, and m set 0.35.

3.5. Implementation details

All of the models are trained using NVIDIA P40 4 GPUs, each GPU with 22GB memory, and accumulating gradients using PyTorch's DistributedDataParallel framework [40]. We use the Adam optimizer with an initial learning rate of 0.001 decreasing by 15% every 10 epochs. A weight decay of 5e-5 is applied. The mixed precision training is used, which allows larger batch sizes and can significantly boost the performance of training models [41].

3.6. Scoring

The trained networks are validated on the development and evaluation sets of the SdSVC. We sample ten 4-second temporal segments at regular intervals from every segments, and repeatedly splicing temporal segment which less than 4-second. The speaker enrollment models are constructed by averaging the corresponding L_2-normalized enrollment embeddings. The verification trials are scored by calculating the pairwise distance between the average enrollment model and ten L_2-normalized test utterance embeddings. The mean of 10 similarities is used as the final score.

3.7. Phrase classification

Task 1 of the SdSVC 2021 is defined as speaker verification in TD mode. It is a twofold verification task in which both speaker and phrase are verified. For a segment of test speech and the target speaker's enrollment data, it has to tell whether the test segment and a specific phrase was spoken by the target speaker. The enrollment and test phrase are drawn from a fixed set of ten sentences that also appear in training data. Therefore, we use phrase classification model to detect whether the test phrase match the phrase label in the enrollment.

The phrase classification model is the same as the speaker embedding network. We use ten temporal convolutional network (TCN) [42] layers as the frame-level feature extractor. Each adjacent two layers with residual connections form a block. Hence, there are 5 blocks. Moreover, we padding the input to ensure that each convolutional layer is causal. The number of dilation parameters in TCN layers used in our experiments are [1, 1, 2, 2, 4, 4, 8, 8, 16, 16], and the kernel size of convolutional layer is set 7. To generate the utterance-level embedding, we use the ASP layer to aggregate the frame-level features. It is followed by a fully connected layer and a softmax classification layer. The cross-entropy loss is used for phrase classification. We utilize 64 band log-mel spectrogram with a 25 ms analysis window and 10 ms overlap. The model is trained on the in-domain data of Task 1. The accuracy of the model on the development set is 99.97%, and the equal error rate (EER) on the test set is 0.1%. We use the model to add a bias of -4 to the trial scores where the test utterance did not to match the enrollment phrase.

3.8. Evaluation

System performance is assessed on the evaluation of SdSVC 2021. The evaluation set is also drawn from DeepMine dataset. For Task 1, the enrollment and test phrases are drawn from a fixed set of ten phrases consisting of five Persian and five English phrases, respectively. Three utterances are used for enrollment. For Task 2, the enrollment data consists of one to several variable-length utterances, and the speech duration for each model is roughly 4 to 180 seconds.

The performance is reported in terms of EER and the normalized detection cost function (MinDCF) as defined in SRE08. This detection cost function for MinDCF is defined as a weighted sum of miss and false alarm error probabilities:

$$C_{Det} = C_{Miss} \times P_{Miss|Target} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm|NonTarget} \times (1 - P_{Target})$$

where $C_{miss} = 10, C_{FalseAlarm} = 1$ and $P_{target} = 0.01$. Based on the parameters, the normalized DCF (DCF_{norm}) will be DCF divide by 0.1 as the best cost that could be obtained without processing the input data.

4. Results and Discussion

4.1. Investigation on speaker embedding extractor

Table 1 illustrates the comparison of different network configurations of speaker embedding extractor. Compared with ResNet and ECAPA-TDNN, Res2Net improves the performance a lot. Based on the Res2Net architecture, we have compared the performance of different integrated modules, namely integrated SE blocks (SE-Res2Net), integrated cardinality dimension (Res2NeXt) and integrated both (SE-Res2NeXt). We observe that integrating SE blocks and cardinality dimension have improved the performance of the original Res2Net architecture, and the SE-Res2Net outperforms others in terms of the MinDCF. Also, comparing ResNet with different layers, we can notice that deeper networks show better performance.

Table 1: *Investigation of different speaker embedding extractor on Task 2, different architectures are described in Section 2.1.*

Architecture	#Params	Development Set		Evaluation Set	
		MinDCF	EER(%)	MinDCF	EER(%)
SE-ResNet-34	8.03M	0.1040	2.20	0.0794	1.76
SE-ResNet-101	13.12M	0.1004	2.19	0.0708	1.58
Res2Net-50	8.29M	0.0634	1.52	0.0612	1.42
SE-Res2Net-50	9.84M	**0.0579**	1.37	**0.0551**	1.26
Res2NeXt-50	8.44M	0.0582	**1.33**	0.0553	**1.22**
SE-Res2NeXt-50	10.01M	0.0593	1.39	0.0577	1.24
ECAPA-TDNN	14.70M	0.0966	2.02	0.0718	1.61
SdSVC x-vector baseline	-	0.3162	8.11	0.4318	10.65

4.2. Investigation on transfer learning

The impact of transfer learning on TD task can be found in Table 2. First of all, we use the best model in Task2, SE-Res2Net-50 as the speaker embedding extractor to compare different transfer learning strategies. We can observe that transfer learning strategies are more effective than training from scratch. Also, the whole model fine-tuning achieves best results. After fine-tuning, all models on the test set have an average improvement of more than 50.3% in EER and 60.6% in MinDCF compared to SdSVC 2021 baseline systems.

4.3. Investigation on fusion systems

Table 3 illustrates the final results of the SdSVC 2021 submitted by IMU&Elevoc. In final submissions, we investigate different fusion methods, and obtain a single model which performance is close to that of multiple model fusion. We fuse multiple models based on the score average and embedding average methods. Task 1 fuses five models after fine-tuning, and Task 2 fuses seven models described in Section 4.1. We can observe that embedding averaging achieves the best performance in our submissions. Then, we distill the fusion model into a single student model which is SE-Res2Net-50 and compare different distillation loss functions. It can be seen that the single model is able to match the performance of the fusion model, and MSE based distillation achieves a better performance than Cosine distance.

Table 2: *Investigation of different transfer learning strategies on Task 1, specific strategies are described in Section 2.2.*

Architecture	Development Set		Evaluation Set	
	MinDCF	EER(%)	MinDCF	EER(%)
SE-Res2Net-50				
— Train from scratch	0.1197	3.14	0.1034	3.01
— Fine-tune whole model	**0.0683**	2.88	**0.0549**	**1.66**
— Fine-tune pooling layer and speaker embedding	0.0898	2.94	0.0709	2.03
Res2Net-50	0.0712	2.97	0.0592	1.71
Res2NeXt-50	0.0694	**2.87**	0.0559	1.68
SE-Res2NeXt-50	0.0709	2.95	0.0590	1.71
ECAPA-TDNN	0.0727	3.01	0.0601	1.77
SdSVC i-vector baseline	0.1629	4.95	0.1468	3.50
SdSVC x-vector baseline	0.5600	10.80	0.5292	9.04

Table 3: *Investigation of different fusion methods on the Task 1 and Task 2 submissions.*

Fusion method	Task 1		Task 2	
	MinDCF	EER(%)	MinDCF	EER(%)
Score averaging	0.0538	1.65	0.0452	1.02
Embedding averaging	**0.0500**	**1.57**	**0.0448**	**1.02**
$\text{Embedding}_{\text{MSE}}$ distillation	0.0505	1.57	0.0450	1.02
$\text{Embedding}_{\text{COS}}$ distillation	0.0511	1.59	0.0460	1.04

5. Conclusions

In this paper, we analyzed the submissions of IMU&Elevoc on Task 1 and Task 2 of the SdSVC 2021. We studied the effectiveness of Res2Net-based frame-level feature extractor and further integrated architectural enhancements to extract more discriminative speaker embeddings. Specifically, we investigated the transfer learning strategy to overcome insufficient Task 1 training dataset, and fine-tune the well-trained whole model of Task 2 based on Task 1 in-domain dataset. Moreover, we successfully obtained a stronger single model based on knowledge distillation method. Finally, our primary submission yield MinDCF of 0.0500 in Task 1 and MinDCF of 0.0448 in Task 2 on the evaluation sets, which are 4th result in the text-dependent task and 6th result in text-independent task.

6. Acknowledgements

We gratefully acknowledge many fruitful discussions with all the members from Elevoc's R&D department. This research work is supported by the National Natural Science Foundation of China (No. 61876214).

7. References

[1] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[2] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason, "I-vector based speaker recognition on short utterances," in *Interspeech*, 2011, pp. 2341–2344.

[3] A. Kanagasundaram, R. Vogt, D. Dean, and S. Sridharan, "Plda based speaker recognition on short utterances," in *Odyssey*, 2012, pp. 28–33.

[4] N. Li, D. Tuo, D. Su, Z. Li, D. Yu, and A. Tencent, "Deep discriminative embeddings for duration robust speaker verification." in *Interspeech*, 2018, pp. 2262–2266.

[5] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification." in *Interspeech*, 2018, pp. 3623–3627.

[6] S. Wang, Z. Huang, Y. Qian, and K. Yu, "Discriminative neural embedding learning for short-duration text-independent speaker verification," *IEEE/ACM TASLP*, vol. 27, no. 11, pp. 1686–1696, 2019.

[7] J.-w. Jung, H.-S. Heo, H.-j. Shim, and H.-J. Yu, "Short utterance compensation in speaker verification via cosine-based teacher-student learning of speaker embeddings," in *ASRU*, 2019, pp. 335–341.

[8] A. Hajavi and A. Etemad, "A deep neural network for short-segment speaker recognition," in *Interspeech*, 2019, pp. 2878–2882.

[9] S. M. Kye, Y. Jung, H. B. Lee, S. J. Hwang, and H. Kim, "Meta-learning for short utterance speaker recognition with imbalance length pairs," in *Interspeech*, 2020, pp. 2982–2986.

[10] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE/ACM TASLP*, vol. 19, no. 4, pp. 788–798, 2010.

[11] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *IEEE SLT*, 2016, pp. 165–170.

[12] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *ICASSP*, 2018, pp. 5329–5333.

[13] P. Kenny, T. Stafylakis, P. Ouellet, V. Gupta, and M. J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition." in *Odyssey*, 2014, pp. 293–298.

[14] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[15] P. Zhang, P. Hu, and X. Zhang, "Deep embedding learning for text-dependent speaker verification," in *Interspeech*, 2020, pp. 3461–3465.

[16] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Odyssey*, 2018, pp. 74–81.

[17] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *ICASSP*, 2016, pp. 5115–5119.

[18] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Interspeech*, 2018, pp. 2252–2256.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[20] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018, pp. 7132–7141.

[21] S. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. H. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Computer Architecture Letters*, no. 01, pp. 1–1, 2019.

[22] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017, pp. 5987–5995.

[23] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Interspeech*, 2020, pp. 3830–3834.

[24] Z. Gao, Y. Song, I. McLoughlin, P. Li, Y. Jiang, and L.-R. Dai, "Improving aggregation and loss function for better embedding learning in end-to-end speaker verification system," in *Interspeech*, 2019, pp. 361–365.

[25] 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 *Interspeech*, 2019, pp. 4045–4049.

[26] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *NIPS*, 2014, pp. 3320–3328.

[27] Z. Allen-Zhu and Y. Li, "Towards understanding ensemble, knowledge distillation and self-distillation in deep learning," *arXiv preprint arXiv:2012.09816*, 2020.

[28] S. Wang, Y. Yang, T. Wang, Y. Qian, and K. Yu, "Knowledge distillation for small foot-print deep speaker embedding," in *ICASSP*, 2019, pp. 6021–6025.

[29] H. Zeinali, H. Sameti, and T. Stafylakis, "Deepmine speech processing database: Text-dependent and independent speaker verification and speech recognition in persian and english." in *Odyssey*, 2018, pp. 386–392.

[30] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Interspeech*, 2017, pp. 2616–2620.

[31] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech*, 2018, pp. 1086–1090.

[32] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206–5210.

[33] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[34] "The webrtc vad project," [Online], 2011, available: https://github.com/wiseman/py-webrtcvad.

[35] K. Tan and D. Wang, "Complex spectral mapping with a convolutional recurrent network for monaural speech enhancement," in *ICASSP*, 2019, pp. 6865–6869.

[36] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[37] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*, 2017, pp. 5220–5224.

[38] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in *CVPR*, 2020, pp. 6398–6407.

[39] P. Zhang, P. Hu, and X. Zhang, "Imu&elevoc system for far-field speaker verification challenge 2020," [Online], 2020, available: http://2020.ffsvc.org/SystemDescription/.

[40] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania *et al.*, "Pytorch distributed: Experiences on accelerating data parallel training," *arXiv preprint arXiv:2006.15704*, 2020.

[41] J. S. Chung, J. Huh, S. Mun, M. Lee, H.-S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," in *Interspeech*, 2020, pp. 2977–2981.

[42] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *CVPR*, 2017, pp. 1003–1012.