INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Deep Speaker Embedding Extraction with Channel-Wise Feature Responses and Additive Supervision Softmax Loss Function

Jianfeng Zhou[1], *Tao Jiang*[2], *Zheng Li*[1], *Lin Li*[1], *Qingyang Hong*[2]

[1]School of Electronic Science and Engineering, Xiamen University, China
[2]School of Information Science and Engineering, Xiamen University, China
{lilin,qyhong}@xmu.edu.cn

Abstract

In speaker verification, the convolutional neural networks (CNN) have been successfully leveraged to achieve a great performance. Most of the models based on CNN primarily focus on learning the distinctive speaker embedding from the horizontal direction (time-axis). However, the feature relationship between channels is usually neglected. In this paper, we firstly aim toward an alternate direction of recalibrating the channel-wise features by introducing the recently proposed "squeeze-and-excitation" (SE) module for image classification. We effectively incorporate the SE blocks in the deep residual networks (ResNet-SE) and demonstrate a slightly improvement on VoxCeleb corpuses. Additionally, we propose a new loss function, namely additive supervision softmax (AS-Softmax), to make full use of the prior knowledge of the mis-classified samples at training stage by imposing more penalty on the mis-classified samples to regularize the training process. The experimental results on VoxCeleb corpuses demonstrate that the proposed loss could further improve the performance of speaker system, especially on the case that the combination of the ResNet-SE and the AS-Softmax.

Index Terms: speaker verification, speaker embedding, AS-Softmax, squeeze-and-excitation block

1. Introduction

The task of speaker verification (SV) is to verify the identity of speaker from a given speech utterance. In the deep learning era, a speaker verification system commonly comprises a front-end speaker embedding extractor which maps variable-length utterances into low-dimensional vectors and a back-end classifier such as probabilistic linear discriminant analysis (PLDA) [1].

In the last decades, the traditional statistical approaches like GMM-UBM [2] and i-vector model [3] have dominated many scenarios of speaker verification until the models based on neural networks shown up. In [4], Snyder et al. used a feed-forward deep neural network to map the variable-length utterances into fixed-dimension speaker embeddings, also known as x-vectors, by calculating the statistics (mean and deviation) along the time-axis, in which each frame contributes equally to the x-vectors. Straight after that, Okabe et al. [5] introduced the widely used attention mechanism in machine translation into the extraction of x-vectors. This method enables the speaker embeddings to be focused on important frames providing importance-weighted standard deviations as well as the weighted means of frame-level features, which help obtain long-term speaker representation with higher discriminative power. Analogous to [5], Chowdhury et al. [6] introduced the different attention mechanisms into the LSTM model to explore different topologies and variants of the attention layer.

However, the mechanisms mentioned above are all focused on the relationship between frames, while the interdependence of the channels (the feature dimension) also contributes to the discriminative representation learning [7, 8]. But rare methods have explored the channel-wise information in speaker verification. Given that, we firstly try to improve the subtle differences learning capability of a speaker embedding system by introducing the squeeze-and-excitation block (SE-Block) [7] into the speaker embedding extractor. The SE-Block is firstly proposed to improve the representational power of a network by enabling it to perform dynamic channel-wise feature recalibration. Specifically, SE-Block is a simplified network that could be inserted in CNN-based network to learn the interdependence between the channels, and produce a set of channel specific weights to emphasize informative features and suppress less useful ones. Given the long-term speaker characteristics derived from standard deviations [5], we have adapted the SE-Block by replacing the global mean pooling in SE-Block with statistics pooling, which is more suitable for the speaker verification task.

Aside from the neural network architecture, the training criteria (loss function) is also a crucial part for learning a discriminative latent representation. Most recent efforts relied on adapting the basic cross entropy loss function, also known as softmax loss, have been explored in speaker verification. In [9, 10], Li et al. and Huang et al. found that a more discriminative speaker embedding could be derived by introducing a margin between the target class and the non-target class into the softmax loss. Similar to [9, 10], the variants of softmax loss, like AM-Softmax and Large Margin Softmax Loss [11, 12, 13], have further improved the discriminability of embedding by enlarging the distance between different speakers from the aspect of angle and margin. Moreover, the end-to-end loss like TE2E [14] and GE2E [15] have been proposed to training the speaker model in an end-to-end fashion. It is worth noting that aforementioned loss functions did not pour much attention on the hard samples, which are beneficial for learning a distinctive representation [16, 17].

Given that, we propose the additive supervision softmax (AS-Softmax) to make full use of prior information that whether a sample have been misclassified during the training process. Specifically, we impose more penalty on the mis-classified samples automatically so that the network could learn the hard-capture features lying on the mis-classified samples. Particularly, we adapt the softmax loss by introducing a adaptive scale term which is controlled by the largest probability that the network predicts the samples should have and the probability that corresponding to the ground truth label. Crucially, the two probabilities could be the same one in the case that the sample is classified correctly.

http://dx.doi.org/10.21437/Interspeech.2019-1704

Table 1: *The topology of ResNet architecture. T is the frame number of each sample and N is the number of speaker in the training set.*

Layer	Output Size	Kernel Size	Stride
Resnet Block1	T×512	5*1	1*1
Resnet Block2	T×512	5*1	1*1
Resnet Block3	T×512	5*1	1*1
Resnet Block4	T×512	7*1	1*1
Resnet Block5	T×512	7*1	1*1
Resnet Block6	T×512	1*1	1*1
Resnet Block7	T×512	1*1	1*1
Resnet Block8	1536	1*1	1*1
Statistics Pooling	3072	-	-
FC1	512	-	-
FC2	256	-	-
Softmax	N	-	-

The paper is organized as follows. Section 2 describes the basic speaker embedding extraction based on ResNet and the ResNet-SE architecture. Section 3 introduces the softmax loss and AS-Softmax. Section 4 and 5 present the experimental setup and the corresponding results based on VoxCeleb corpuses respectively. Section 6 summarizes this paper.

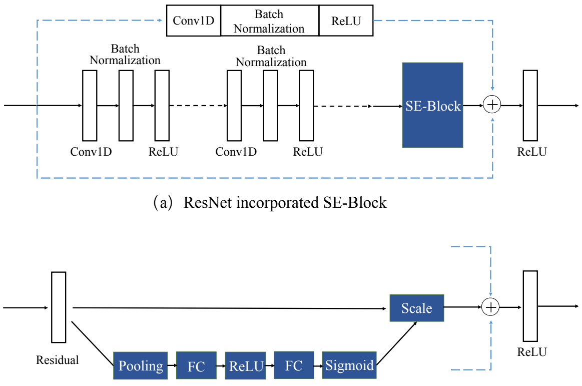
2. Deep speaker embedding extractor

2.1. Speaker embedding extraction based on ResNet

ResNet-based architecture has been widely used as a speaker embedding extractor in speaker verification tasks [18, 19]. In this work, we realized the speaker embedding extractor based on ResNet and we replace the convolutional layer in ResNet [20] with 1-dimentional convolutional layer (Conv1D), which is more suitable for processing audio signal. The whole architecture includes eight ResNet blocks that operate on frame level, a statistics pooling layer that calculates the mean and standard deviation of each samples along the time-axis, and finally two fully-connected (FC) layers and a softmax layer on segment level. The details of the ResNet architecture are provided in Table 1. Specifically, the parameters shown in each row of Table 1 are for all the Conv1D layers in the corresponding ResNet block.

2.2. ResNet with squeeze-and-excitation block

In this work, we incorporate the SE-Block in the ResNet architecture to boost speaker discriminability learning capability of the embedding extractor. The way how the SE-Block is incorporated in ResNet is shown in Figure 1. (a) and the details are shown in Figure 1. (b). The output of the residual block $O \in R^{T*C}$ flows through the SE-Block before the skip connection, where T denotes the frame number of the samples and C denotes the channel number of the residual block. With regard to the details of SE-Block, a pooling layer aggregates along the time-axis to generate channel-wise statistics (mean and standard deviation, which will be concatenated and fed to the next layers). Then, two fully-connected layers with $\frac{C}{r}$ nodes and C nodes respectively are used to capture channel-wise interdependence based on the channel-wise statistics. The parameter r is a reduction ratio for controlling the computational cost of SE-Block (we set r as 16 in our implementation, which is a parameter depended on the balance of performance and computational cost [7]). Finally, a scale operator conducts the channel-wise multiplication to reinforce informative features and suppress less effective ones.



(a) ResNet incorporated SE-Block

(b) Details of SE-Block

Figure 1: *The schematic of the ResNet-SE module (a) and the details of ResNet-SE module (b). The dotted line means the path are alternative, which depends on whether the dimension of x equals to that of the SE block output. When they are the same, the upper one would be chosen, otherwise the other.*

With this approach, the information that is conducive to help the neural network improve the speaker discriminability is emphasized and the other is overshadowed, so that the speaker embeddings learned from the architecture are more concentrated on speaker characteristics.

3. Additive supervision softmax loss

3.1. Softmax-based loss function

As a common loss for classification tasks, the cross entropy loss function, also known as softmax loss, indicates the distance between what the model believes the output distribution should be, and what the original distribution really is, which is almost a default setting in speaker verification tasks since the neural network methods have sprung up. The softmax-based loss could be formulated as follow:

$$L_{softmax} = -\frac{1}{M}\sum_{i=1}^{M} log\left[y_i^*\left(x\right)\right] \tag{1}$$

where the M is the batch size and $y_i^*(x)$ is the softmax output of the ith sample corresponding to the ground truth label. As for optimization, the softmax-based loss is ordinarily used to train the network by $\min_{\theta} L_{softmax}$ (θ represents the parameters of the network) to increase the probability of assigning the speaker embedding to its corresponding label.

3.2. Additive supervision softmax loss

It is intuitive that the mis-classified samples contribute more to the improvement of the identity discriminability [16]. Given this, we proposed the additive supervision softmax loss (AS-softmax) to make full use of the prior knowledge of the mis-classified samples. In contrast to the original softmax loss, the AS-softmax takes the prior information of whether a sample has been classified correctly into account, and then imposes more penalty on the mis-classified samples to regularize the network encoding the hard-capture speaker information. The formulation of the AS-softmax is shown in (2),

Table 2: *EER (%) of the SV systems evaluated on the original test set of VoxCeleb1. VoxCeleb2* is a subset of VoxCeleb2. Due to the resource and time constrants, we randomly sample one utterence from every video of the same speaker to construct a subset of Voxceleb2 for training(about 140,000 utterences).*

-	Front-end	Loss	Back-end	Training set	Toolkit	EER
Nagrani et al.[21]	VGG-M	Softmax	-	VoxCeleb1	-	7.80
Cai et al.[22]	ResNet-34	A-Softmax	PLDA	VoxCeleb1	Pytorch	4.40
Hajibabaei et al.[12]	ResNet-20	AM-Softmax	-	VoxCeleb1	Caffe	4.30
Okabe et al. [5]	TDNN (x-vector)	Softmax	-	VoxCeleb1	Kaldi	3.85
Ours	ResNet-18	Softmax	PLDA	VoxCeleb1	Tensorflow	4.58
	ResNet-18-SE	Softmax		VoxCeleb1		4.45
	ResNet-18	AS-Softmax		VoxCeleb1		4.43
	ResNet-18-SE	AS-Softmax		VoxCeleb1		4.06
	ResNet-34-SE	AS-Softmax		VoxCeleb1		**3.52**
Chung et al.[19]	ResNet-50	Softmax + Contrastive	-	VoxCeleb2	-	3.95
Xie et al.[23]	Thin ResNet-34	Softmax	-	VoxCeleb2	-	3.22
Ours	ResNet-18	Softmax	PLDA	VoxCeleb2*	Tensorflow	4.61
	ResNet-18-SE	Softmax		VoxCeleb2*		4.15
	ResNet-18	AS-Softmax		VoxCeleb2*		4.35
	ResNet-18-SE	AS-Softmax		VoxCeleb2*		3.79
	ResNet-34-SE	AS-Softmax		VoxCeleb2		**3.10**

$$\begin{aligned} L_{AS} = &-\frac{1}{2M}\sum_{i=1}^{M}(log[y_i^*(x)] \\ &+ \frac{(log[y_i^*(x)])^2}{log[\max\{y_i^1(x), y_i^2(x), ..., y_i^N(x)\}]+\delta}) \end{aligned} \tag{2}$$

where N is the number of speakers in training set and δ is a minor negative number that makes sure the denominator is not zero. $y_i^k(k = 1, 2, ..., N)$ means the kth softmax output corresponding to the label k. For the easy explanation, we make assumption as follows:

$$V^S = log[y_i^*(x)] \tag{3}$$

$$V^{AS} = log[\max\{y_i^1(x), y_i^2(x), ..., y_i^N(x)\}] \tag{4}$$

Then the equation (2) could be reformulated as follow:

$$\begin{aligned} L_{AS} &= -\frac{1}{2M}\sum_{i=1}^{M}(V^S + \frac{(V^S)^2}{V^{AS}+\delta}) \\ &\approx -\frac{1}{M}\sum_{i=1}^{M}V^S(\frac{1+\frac{V^S}{V^{AS}}}{2}) \end{aligned} \tag{5}$$

Obviously, when the samples have been classified correctly, the V^{AS} equals to V^S, which means the L_{AS} approximately equals to $L_{softmax}$. But for the mis-classified samples, the absolute value of V^{AS} is less than that of V^S, so the L_{AS} is greater than $L_{softmax}$. Besides, the larger the probability that the network misclassified the samples, the larger the penalty is. During the training process, the AS-softmax imposes more penalty on the mis-classified samples which helps capture the subtle differences lying on mis-classified samples and improve the speaker discriminability.

4. Experiments

4.1. Datasets and experimental settings

In our experiments, we verify the effectiveness of the proposed methods on the VoxCeleb1 [21] and VoxCeleb2 [19] datasets, the ’dev’ partition of which include 1211 and 5994 celebrities respectively. And we augmented the ’dev’ parition of VoxCelecb1 for training by adding the noise data from Musan [24] and reverberation [25] to multiply the amount of training data using the method proposed in [26]. For the raw feature extraction, all audios were converted to the cepstral features of 23-dimensional MFCC with a frame-length of 25ms and a frame shift of 10ms. Then, a frame-level energy-based voice activity detector (VAD) selection was conducted to the features. This was followed by local cepstral mean and variance normalization (CMVN) over a 3-second sliding window. All operations of feature extraction and the verification stage were based on Kaldi toolkit [27]. Additionally, our network implementation and training was realized on the Tensorflow toolkit [28].

4.2. Other details

For the training, the audio samples are sliced into pieces of segments with 200 frames. Then the segments with the size 200×23 are fed to the network for forward propagation. Finally, the loss function takes the output of the network to calculate gradients for the backpropagation. In the verification stage, the output layer of the network is discarded and the speaker embeddings are extracted at layer FC2. Meanwhile, global mean subtraction, length normalization and linear discriminant analysis (LDA) transformation are applied to speaker embeddings and then PLDA is used for scoring.

In our experiments, Adam optimizer [29] with a learning rate of 0.001 was used for the backpropagation and 50 percent probability of dropout [30] was applied to the first hidden layer.

5. Results

5.1. Verification on original test set

The proposed methods are evaluated on the original test data of the VoxCeleb1 which includes about 37,720 trials from 40 speakers. Though we slice the samples in the training process, we use the full length of each sample for testing. The performance measure is equal error rate (EER) and the results are shown in Table 2.

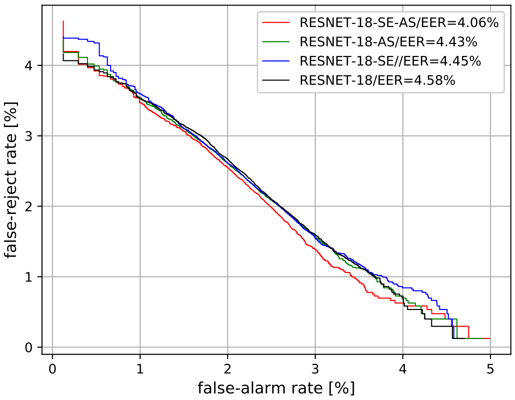


Figure 2: *DET curves. Results of our four methods training on VoxCeleb1 are displayed.*

Table 3: *EER (%) of the SV systems evaluated on the extended and hard test sets of VoxCeleb1.*

Models	Test set	EER(%)
ResNet-50 [19]	VoxCeleb1-E	4.42
ResNet-34-SE(AS)	VoxCeleb1-E	**3.38**
ResNet-50 [19]	VoxCeleb1-H	7.33
ResNet-34-SE(AS)	VoxCeleb1-H	**5.93**

On the comparison of ResNet-18 and ResNet-18-SE shown in Table 2, we observe that the ResNet-18-SE outperforms the ResNet-18 with both VoxCeleb1 and Voxceleb2* training set. The improvement of the performance benefits from the recalibration of channel-wise features, which help capture the subtle differences between identities. Then, to verify the effectiveness of the AS-Softmax, we built the SV systems based on ResNet-18 or ResNet-18-SE with Softmax function and the proposed AS-Softmax function, respectively. It is obviously that the SV systems with AS-Softmax function achieved lower EER values compared with those with Softmax function, specifically, 3.28% relative reduction for ResNet-18 and 9.61% for ResNet-18-SE on VoxCeleb1 while 5.64% for ResNet-18 and 8.67% for ResNet-18-SE on VoxCeleb2*, respectively. It is a remarkable fact that the larger relative reduction could be obtained by the combination of the SE-Block and the AS-Softmax function. That is beneficial from combination of the subtle differences learning ability of the SE-Block and the constraint that AS-Softmax has imposed on mis-classified samples. The greater constraint means a more powerful model are required to encode the hard-capture features. Moreover, we also executed the proposed method using a deeper network ResNet-34-SE, the result of which (3.52% in EER), to the best of our knowledge, has outperformed the public counterparts trained on VoxCeleb1 and evaluated on the original test set of VoxCeleb1. It is worth nothing that ResNet-18-SE just costs slight additional computational burden in exchange for the performance gain. The ResNet-18 model is with 17.6M parameters and ResNet-18-SE model is with 18.4M parameters approximately.

For a more comprehensive evaluation, the detection error trade-off (DET) curves of the proposed systems (Ours) are displayed in Figure 2. Besides, the convergent tendency of the training accuracy is shown in Figure 3. It is observed that SV system with ResNet-SE and the AS-Softmax could not only achieve the best performance, but also accelerate the optimization process deducing from the tendency of training accuracy shown in Figure 3.

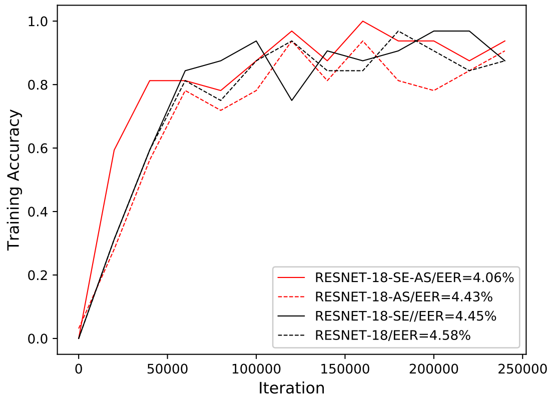


Figure 3: *Training accuracy of our four systems training on VoxCeleb1.*

5.2. Verification on the extended and hard test set

Apart from the original test set, we also conducted our experiments on the new test sets, the extended and hard test sets (VoxCeleb1-E and VoxCeleb1-H) [19], both of which are sampled from the Voxceleb1 dataset. Specifically, in order to address the possibility of overfitting the small original dataset [19], the extended test set consisting of 581,480 random pairs has been released to verify the good generalized performance of speaker systems. And the hard test set consists of 552,536 pairs from the same nationality and gender. The results of ResNet-34-SE model with AS-Softmax are given in Table 3, all of which outperform the results shown in [19].

6. Conclusion

In this paper, we have firstly explored the potential advantage of ResNet-SE in learning the more subtle information lying in audios by reinforcing the useful information and weakening the others for speaker verification tasks. Results showed that ResNet-SE could outperform the basic ResNet. Besides, the proposed AS-Softmax, which utilizes the additive supervision of the mis-classified samples in training process by imposing more penalty on the samples, has shown its great effectiveness. Especially, the proposed speaker verification system with the ResNet-SE and AS-Softmax function has achieved the outstanding performance on test sets of the VoxCeleb1.

Since the AS-Softmax and other softmax-based loss such as A-Softmax, AM-Softmax are concentrated on the different aspect, we would like to explore the possibility of combining the AS-Softmax with A-Softmax or AM-Softmax to further improve the performance in the future.

7. Acknowledgement

This work was supported by the National Natural Science Foundation of China (Grant No.61876160).

8. References

[1] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[2] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[3] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[4] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification." in *Interspeech*, 2017, pp. 999–1003.

[5] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," *arXiv preprint arXiv:1803.10963*, 2018.

[6] F. Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," *arXiv preprint arXiv:1710.10470*, 2017.

[7] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[8] S. Gong, Y. Shi, and A. K. Jain, "Video face recognition: Component-wise feature aggregation network (c-fan)," *arXiv preprint arXiv:1902.07327*, 2019.

[9] Y. Li, F. Gao, Z. Ou, and J. Sun, "Angular softmax loss for end-to-end speaker verification," *arXiv preprint arXiv:1806.03464*, 2018.

[10] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification," *Proc. Interspeech, Hyderabad*, 2018.

[11] G. Bhattacharya, J. Alam, and P. Kenny, "Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training," *arXiv preprint arXiv:1811.03055*, 2018.

[12] M. Hajibabaei and D. Dai, "Unified hypersphere embedding for speaker recognition," *arXiv preprint arXiv:1807.08312*, 2018.

[13] G. Bhattacharya, J. Monteiro, J. Alam, and P. Kenny, "Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification," *arXiv preprint arXiv:1811.03063*, 2018.

[14] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[15] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[16] X. Wang, S. Wang, S. Zhang, T. Fu, H. Shi, and T. Mei, "Support vector guided softmax loss for face recognition," *arXiv preprint arXiv:1812.11317*, 2018.

[17] L. Xu, H. Sun, and Y. Liu, "Learning with batch-wise optimal transport loss for 3d shape recognition," *arXiv preprint arXiv:1903.08923*, 2019.

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

[19] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[20] 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.

[21] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[22] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," *arXiv preprint arXiv:1804.05160*, 2018.

[23] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Xie w, nagrani a, chung j s, et al. utterance-level aggregation for speaker recognition in the wild," *arXiv preprint arXiv:1902.10107*, 2019.

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

[25] 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 *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

[26] 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.

[27] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.

[28] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.