The DKU-SMIIP System for NIST 2018 Speaker Recognition Evaluation

Danwei Cai[1], *Weicheng Cai*[1,2], *Ming Li*[1]

[1]Data Science Research Center, Duke Kunshan University, Kunshan, China
[2]School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China

ming.li369@dukekunshan.edu.cn

Abstract

In this paper, we present the system submission for the NIST 2018 Speaker Recognition Evaluation by DKU Speech and Multi-Modal Intelligent Information Processing (SMIIP) Lab. We explore various kinds of state-of-the-art front-end extractors as well as back-end modeling for text-independent speaker verifications. Our submitted primary systems employ multiple state-of-the-art front-end extractors, including the MFCC i-vector, the DNN tandem i-vector, the TDNN x-vector, and the deep ResNet. After speaker embedding is extracted, we exploit several kinds of back-end modeling to perform variability compensation and domain adaptation for mismatch training and testing conditions. The final submitted system on the fixed condition obtains actual detection cost of 0.392 and 0.494 on CMN2 and VAST evaluation data respectively. After the official evaluation, we further extend our experiments by investigating multiple encoding layer designs and loss functions for the deep ResNet system.

Index Terms: speaker verification, NIST SRE 2018, deep embedding, ResNet, domain mismatch

1. Introduction

Since 1996, the US National Institute of Standards and Technology (NIST) has been conducting speaker recognition evaluations (SRE) to explore promising new ideas and measure the performance the state-of-the-art speaker recognition systems [1]. NIST SRE 2018 focus on text-independent speaker verification (TISV) and contains two testing tasks:

- CMN2: Similar to SRE16, the development and evaluation data of CMN2 task are non-English conversational telephone speech collected with various telephone channels. The duration of the test segments ranges 10-60 seconds with enrollment limited to 1 minute. The channel-, language- and duration-mismatches impose great challenges to the CMN2 task.
- VAST: the VAST task of SRE18 contains English audio extracted from YouTube videos that vary in duration from a few seconds to several minutes. The audios are obscured in complex environmental settings including reverberation and noises. Each audio recording may contain speech from multiple talkers. Manually produced diarization labels are provided for the enrollment audio but not for the test audio.

There are two major categories of methods to extract the speaker embedding for TISV. The first comprises stacking self-contained algorithmic components, and the representative is the classical i-vector approach [2]. The i-vector extractor can be trained using either the acoustic-level features or phoneme discriminant features extracted from the additional acoustic model [3, 4].

The second category relies on the model trained by a downstream procedure through a deep neural network (DNN). The representative is the x-vector system [5] based on time delay neural network (TDNN). Recently, Cai *et al.* have achieved state-of-the-art performance by extracting speaker embeddings from a deep residual neural network (ResNet) [6]. Based on the deep ResNet framework, various kinds of encoding layers like learnable dictionary encoding (LDE) layer [7] and NetVLAD layer [8] as well as loss functions like center loss [9, 6] and angular softmax loss (A-softmax) [10, 6] are explored and improve the performance significantly.

Our developed systems consist of multiple state-of-the-art front-end extractors, including MFCC i-vector, DNN tandem i-vector, TDNN x-vector, and deep ResNet. After speaker embeddings are extracted, we investigate several back-ends modeling and score normalization methods. Different components including variabilities compensation, domain adaptation, in-domain whitening and Gaussian Probability Linear Discriminant Analysis (PLDA) are used for scoring. Also, cosine similarity is used as an alternative scoring method for ResNet based systems.

This paper is organized as follows: Section 2 describes the details of our submitted system. Section 3 illustrates the submission performance of both the single and fusion systems. Some of our post-evaluation experiments and analysis are presented in section 4. Conclusions are drawn in section 5.

2. System descriptions

2.1. Front-end extractor

2.1.1. MFCC i-vector

The MFCC i-vector system is developed by adapting the Kaldi SRE16 recipe. 20-dimensional MFCC is augmented with their delta and double delta coefficients, making 60-dimensional feature vectors. A simple energy-based voice activity detector (VAD) is used. A short-time cepstral mean subtraction (CMS) is applied on the over a 3-second sliding window. A 2048-components full covariance Gaussian Mixture Model-Universal Background Model (UBM) is trained, along with a 600-dimensional i-vector extractor [2].

2.1.2. DNN tandem i-vector

The DNN tandem i-vector system uses the tandem feature, which has a phonetic-aware feature concatenated with the acoustic MFCC feature [3, 4]. For phonetic feature extraction, we employed a DNN acoustic model, trained on Fisher dataset, with 5621 tied tri-phone states to get the frame-level phoneme

This research was funded in part by the National Natural Science Foundation of China (61773413), Natural Science Foundation of Guangzhou City (201707010363), Six Talent Peaks project in Jiangsu Province (JY-074), Science and Technology Program of Guangzhou City (201903010040).

posterior probability (PPP). After logarithm and Principal Component Analysis (PCA), the resulted 52-dimensional phonetic aware features are fused with the 60-dimensional MFCC at the feature level to get the 112-dimensional tandem feature. After the tandem feature is extracted, the subsequent UBM and factor analysis setup is exactly the same as in the MFCC i-vector recipe.

2.1.3. TDNN x-vector

The x-vector [5] system is developed by adapting the Kaldi SRE16 recipe. For the x-vector extractor, a DNN is trained to discriminate speakers in the training set. The first five timed delayed layers operate at frame-level. Then a temporal statistics pooling layer is employed to compute the mean and standard deviation over all frames for an input segment. The resulted segment-level representation is then fed into two fully connected layers to classify the speakers in the training set. After training, speaker embeddings are extracted from the 512-dimensional affine component of the first fully connected layer.

2.1.4. Deep ResNet

For feature extraction, each audio is converted to 64-dimensional log Mel-filterbank energies. CMS and VAD operation is performed the same as in the MFCC i-vector.

We follow the deep ResNet system as described in [6, 11, 12], and we increase the widths (number of channels) of the residual blocks from {16, 32, 64, 128} to {32, 64, 128, 256}. The network structure contains three main components: a front-end ResNet, a pooling layer, and a feed-forward network. Given a feature sequence of size $D \times L$, the ResNet learns three-dimensional feature descriptions of shape $C \times H \times W$, where C denotes the number of channels, H and W denotes the height and width of the feature maps. To get the single utterance-level representation, we adopt a global average pooling (GAP) layer, which accumulates mean statistics for each feature map. Given feature maps $\mathbf{F} \in \mathbb{R}^{C \times H \times W}$, the output of GAP is defined as:

$$v_i = \frac{1}{H \times W} \sum_{j=1}^{j=H} \sum_{k=1}^{k=W} \mathbf{F}_{i,j,k} \tag{1}$$

Therefore, we get fixed-dimensional utterance-level representation $\mathbf{V} = [v_1, v_2, \cdots, v_C]$ for a variable-length utterance. We further process the utterance-level representation through a classifier with two fully-connected layers. In the output layer, each unit is represented as a target speaker identity. All the components in the pipeline are jointly learned in an end-to-end manner with a unified loss function.

We design a variable-length data loader to generate mini-batch training samples on the fly. For each training step, a random integer L which indicates the frame-length is created. The data loader provides dynamic mini-batch data of shape $B \times D \times L$ before each training step, where B denotes the batch size and D denotes the feature dimension. Therefore, the length of the mini-batch training samples is a batch-wise variable number.

After training, the 256-dimensional utterance-level speaker embedding is extracted after the penultimate layer of the neural network for the given utterance. In the testing stage, the full-length feature sequence is directly fed into the network, without any truncate or padding operation. More detailed network setup and training config can refer to [6].

2.2. Back-end modeling

We trained different back-ends for CMN2 task and VAST task. Details are described as follows.

2.2.1. Variabilities compensation

For variabilities compensation, we use either Linear Discriminant Analysis (LDA) or Locality Sensitive Discriminant Analysis (LSDA) [13, 14] to select the most speaker relevant feature subset and reduce the variabilities irrelevant to the speaker.

Before applying LDA or LSDA, the speaker discriminant features are mean centered. It is noticed that as the training data contains different dataset such as Voxceleb, Call My Net (CMN) and NIST SRE, we apply mean subtraction separately for each dataset to diminish the inter-dataset variabilities.

2.2.2. Domain adaptation

We use Correlation Alignment (CORAL) [15, 16] to align the distributions of out-of-domain and in-domain features in an unsupervised way by aligning second-order statistics, i.e., covariance. CORAL minimize the distance between the covariance of the out-of-domain and in-domain features, and a linear transformation $\mathbf{A}$ to the source features and the Frobenius norm is used as a matrix distance metric.

In CMN2 condition, we use SRE 18 unlabeled data as the target-domain data and the PLDA training data as the source-domain data, and the linear transformation $\mathbf{A}$ is estimated by CORAL algorithm on these two data.

2.2.3. In-domain whitening

To further reduce the variabilities between training data and testing data, in-domain whitening (inW) is applied before PLDA. In-domain whitening calculates the mean and covariance of the in-domain data and applies them to whiten the test data. The whitening transforms are estimated with the SRE 18 unlabeled set and the Speaker In The Wild (SITW) dataset [17] for CMN2 and VAST respectively.

2.2.4. Gaussian PLDA

After whitening, unit-length normalization is applied to the speaker discriminant features. The Gaussian PLDA [18] model with a full covariance residual noise term and a full-rank eigenvoice subspace is then trained for scoring.

2.2.5. Cosine similarity

We use cosine similarity as an alternative back-end for the ResNet based systems. The scores of any given enrolment-test pair are calculated as the cosine similarity of the two features.

2.3. Score normalization

After scoring, all trial results are subject to score normalization. We utilize Adaptive Symmetric Score Normalization (AS-Norm) in our systems. Two variants of AS-Norm associate with the adaptive cohort selection are investigated. The details can be found in [19].

The selection of the score normalization corpus is tuned to be optimal for the development data. For CMN2 condition, the score normalization cohort is SRE 18 unlabeled and SRE 16 unlabeled data. For VAST condition, the score normalization cohort is the SITW dataset.

Table 1: *NIST SRE 2018 CMN2 results for our submitted system on the fixed condition (EER[%] / minC / [actC])*

Front-end	Back-end	Score Norm	Development	Evaluation
MFCC i-vector	LDA + inW + PLDA	-	11.00 / 0.681	12.21 / 0.745
	LDA + CORAL + inW + PLDA	-	11.25 / 0.638	12.82 / 0.708
	LDA + PLDA	AS-Norm2	10.64 / 0.603	**11.93 / 0.667**
	LDA + CORAL + PLDA	AS-Norm2	11.48 / 0.606	12.77 / 0.699
	LSDA + CORAL + PLDA	-	13.27 / 0.624	12.89 / 0.727
	LSDA + CORAL + PLDA	AS-Norm1	**12.95 / 0.581**	12.93 / 0.686
	LSDA + CORAL + PLDA	AS-Norm2	12.18 / 0.590	12.64 / 0.685
DNN tandem i-vector	LDA + inW + PLDA	-	10.64 / 0.669	11.32 / 0.691
	LDA + CORAL + inW + PLDA	-	09.81 / 0.546	10.83 / 0.614
	LDA + PLDA	AS-Norm1	10.04 / 0.549	11.00 / 0.603
	LDA + CORAL + PLDA	AS-Norm1	10.15 / 0.524	10.97 / 0.603
	LSDA + CORAL + PLDA	-	09.03 / 0.533	09.91 / 0.606
	LSDA + CORAL + PLDA	AS-Norm1	**08.85 / 0.494**	09.68 / 0.555
	LSDA + CORAL + PLDA	AS-Norm2	08.10 / 0.495	**09.55 / 0.549**
x-vector	LDA + inW + PLDA	-	07.77 / 0.587	08.89 / 0.587
	LDA + CORAL + inW + PLDA	-	07.09 / 0.469	07.43 / 0.518
	LDA + PLDA	AS-Norm2	07.17 / 0.479	**07.68 / 0.492**
	LDA + CORAL + PLDA	AS-Norm2	**07.32 / 0.419**	07.50 / 0.504
ResNet v1	LDA + CORAL + PLDA	-	07.99 / 0.501	**08.18 / 0.540**
	LDA + CORAL + PLDA	AS-Norm1	**08.60 / 0.475**	08.17 / 0.546
	LDA + CORAL + PLDA	AS-Norm2	**08.23 / 0.475**	08.13 / 0.549
	cosine similarity	AS-Norm1	12.15 / 0.576	12.20 / 0.632
ResNet v2	LDA + CORAL + PLDA	-	09.71 / 0.637	08.85 / 0.616
	LDA + CORAL + PLDA	AS-Norm1	**09.56 / 0.528**	08.53 / 0.554
	LDA + CORAL + PLDA	AS-Norm2	08.92 / 0.535	**08.19 / 0.546**
	cosine similarity	AS-Norm1	13.28 / 0.571	13.12 / 0.636
Fusion			**5.21 / 0.324 / 0.329**	**5.53 / 0.392 / 0.409**

2.4. System fusion and calibration

All the subsystems were fused and calibrated using the BOSARIS toolkit [20] which learn a scale and a bias for each system by logistic regression approach. The final fusion is an equal-weighted sum of the systems after applying the scale and the bias. Fusion applies to CMN2 and VAST tasks separately.

3. Submitted system performance

3.1. Data preparation

The training data includes SRE04-16, MIXER 6, Switchboard, VoxCeleb1 [21] and VoxCeleb2 [22], resulting 14,467 speakers altogether.

Data augmentation is utilized for both x-vector and ResNet systems. We adopt the same data augmentation strategy as the Kaldi x-vector recipe. It employs additive noises and reverberation. Reverberation involves convolving room impulse responses (RIR) with audio and the simulated RIRs described by Ko et al. in [23] are used. For additive noise, the MUSAN dataset [24] is used.

For the MFCC i-vector, DNN Tandem i-vector, and ResNet v1 system, the whole 14,467 speakers are used for training. For the x-vector system, since short-duration utterances and speakers with too little utterances are dropped, 12,459 speakers are used for training. We also train the ResNet v2 system on only the VoxCeleb1 and Voxceleb2 corpus, which includes 7,245 speakers.

3.2. System performance

In table 1, we present the Equal Error Rate (EER) and SRE18 primary Detection Cost Function (DCF) of our submitted systems in the fixed condition for the CMN2 task. Some observations come from the results. First of all, although the DNN tandem i-vector system utilizes a DNN acoustic model trained on English corpus to extract phonetic aware features, it still shows competitive performance on the language-mismatch testing condition after applying the backend with variabilities compensation, domain adaptation, and score normalization. Also, LSDA obtains approximate 5% relative performance gain compared to the LDA algorithm. Moreover, although stated in [6], a simple cosine can achieve good performance in the ResNet based systems, the back-end methods including variabilities compensation, domain adaptation, in-domain whitening, and score normalization are beneficial in training-testing mismatch conditions. The last observation is that different combination of the mismatch reduction algorithms can further improve the system performance, and these systems are also complementary to each other.

Table 2 presents the results of our submitted systems in the fixed condition on VAST task. It is surprising to see that the DNN tandem i-vector system outperform the end-to-end x-vector and ResNet based system in terms of minC on the evaluation set. The reason is that the score normalization setting (AS-Norm2 with 200 cohort scores) used for the x-vector system, which is selected to be optimal for the small trials of the VAST development set (37 utterances, 270 trials), is not well

Table 2: *NIST SRE 2018 VAST results for our submitted system on the fixed condition (EER[%] / minC / [actC])*

Front-end	Back-end	Score Norm	Development	Evaluation
MFCC i-vector	LDA + PLDA	AS-Norm2	11.11 / 0.568	17.46 / 0.590
	LSDA + PLDA	-	16.05 / 0.597	19.90 / 0.613
DNN tandem i-vector	LDA + PLDA	AS-Norm1	10.70 / 0.370	14.98 / 0.521
	LSDA + PLDA	-	10.29 / 0.407	**18.41 / 0.515**
x-vector	LDA + PLDA	AS-Norm2	**08.64 / 0.296**	13.33 / 0.533
ResNet v1	LDA + PLDA	AS-Norm2	08.64 / 0.490	13.97 / 0.609
	cosine similarity	AS-Norm1	16.05 / 0.667	16.19 / 0.771
	cosine similarity	AS-Norm2	14.81 / 0.630	16.83 / 0.739
ResNet v2	LDA + PLDA	-	13.17 / 0.667	16.19 / 0.771
Fusion			**7.41 / 0.259 / 0.259**	**12.77 / 0.494 / 0.515**

Table 3: *CMN2 and VAST results for our post evaluation systems on the fixed condition (EER[%] / minC / [actC])*

ID	Encoding Layer	Loss	CMN2		VAST	
			Development	Evaluation	Development	Evaluation
1	GAP (mean)	Softmax	7.85 / 0.501	7.43 / 0.557	11.93 / 0.486	14.93 / 0.542
2	GAP (mean+std)	Softmax	7.03 / 0.481	7.12 / 0.489	**10.29 / 0.333**	11.75 / 0.480
3	LDE	Softmax	**7.50 / 0.408**	7.17 / 0.503	09.47 / 0.449	14.33 / 0.523
4	GAP (mean)	A-softmax	6.03 / 0.420	6.61 / 0.474	10.29 / 0.370	13.06 / 0.494
5	GAP (mean+std)	A-softmax	5.94 / 0.418	**6.14 / 0.463**	**03.70 / 0.300**	**10.79 / 0.423**
6	LDE	A-softmax	**6.03 / 0.354**	**6.20 / 0.430**	08.23 / 0.407	**13.97 / 0.457**
Fusion: submitted system			5.21 / 0.324 / 0.329	5.53 / 0.392 / 0.409	7.41 / 0.259 / 0.259	12.77 / 0.494 / 0.515
Fusion: post-evaluation systems 1-6			5.32 / 0.354 / 0.368	5.36 / 0.399 / 0.409	3.70 / 0.189 / 0.374	10.94 / 0.412 / 0.421
Fusion: submission + post-evaluation			**4.27 / 0.275 / 0.279**	**4.81 / 0.365 / 0.386**	0.00 / 0.000 / 0.000	**10.74 / 0.401 / 0.406**

tuned for the evaluation set. Indeed, the minC reaches 0.5 when we use AS-Norm1 with 100 cohort scores on x-vector system.

4. Post evaluation

After the evaluation, we further extend our experiments by investigating more encoding layer designs and loss functions for our deep ResNet system.

First, following the x-vector system that accumulates mean and standard deviation statistics in pooling layer, we boost the GAP layer by adding the global standard deviation statistics of the learned feature maps. Specifically, 256-dimensional mean statistics, as well as 256-dimensional standard deviation statistics, are concatenated together to form the utterance-level representation. Furthermore, we replace the GAP layer with LDE layer. It learns a dictionary with the centers of several clusters and encodes the variable-length inputs into a single utterance-level supervector[6, 7]. In our experiments, the number of dictionary components is set to 64.

Second, regarding that the superiority of A-softmax for speaker recognition has been shown in [6], we use A-softmax loss to replace the basic softmax loss. In our experiment, we use the angular margin $m = 4$. When training the network with A-Softmax loss, we use an annealing optimization strategy, which supervises the network from the original softmax loss gradually to A-softmax loss [10].

Table 3 shows the post evaluation results of deep ResNet system on NIST SRE 2018 fixed condition. The best back-end setting and score normalization for the CMN2 systems are tuned on the development set and then apply to the evaluation set, while the VAST system is tuned directly on the evaluation set since the development set for VAST is small, and the results on such small development set are unreliable to some extent.

From table 3, we can see that the LDE layer shows great potential in front-end modeling under language-mismatch condition. Also, GAP layer integrated with standard deviation statistics obtains better performance than the basic GAP layer in both CMN2 and VAST tasks and achieves the best performance in the VAST task. As demonstrated in [6, 10], the system with A-softmax loss always shows an apparent reduction in both EER and DCF compared to the system with softmax loss.

We also provide the fusion results of our post-evaluation systems in table 3. The final fusion with submitted systems and post-evaluation systems obtains 17.8%, 7.4%, and 23.2% relative improvements in terms of minC over the CMN2 development, CMN2 evaluation, and VAST evaluation respectively.

5. Conclusions

In this paper, our submitted DKU-SMIIP system for the NIST SRE 2018 is described. Various kinds of state-of-the-art speaker embedding extractors are explored. We also utilize variabilities compensation, domain adaptation, in-domain whitening, and score normalization algorithms to reduce the mismatch condition between training and testing data. The experimental results show the potential of deep ResNet for large-scale TISV and demonstrate the significance of LDE layer and A-softmax loss.

6. References

[1] National Institute of Standards and Technology, "NIST 2018 Speaker Recognition Evaluation Plan," 2018. [Online]. Available: https://www.nist.gov/sites/default/files/documents/2018/08/17/sre18_eval_plan_2018-05-31_v6.pdf

[2] 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.

[3] M. Li and W. Liu, "Speaker Verification and Spoken Language Identification using a Generalized i-vector Framework with Phonetic Tokenizations and Tandem Features," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014, pp. 1120–1124.

[4] M. Li, L. Liu, W. Cai, and W. Liu, "Generalized i-vector Representation with Phonetic Tokenizations and Tandem Features for both Text Independent and Text Dependent Speaker Verification," *Journal of Signal Processing Systems*, vol. 82, no. 2, pp. 207–215, 2016.

[5] 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)*, 2018, pp. 5329–5333.

[6] 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: The Speaker and Language Recognition Workshop*, 2018, pp. 74–81.

[7] W. Cai, Z. Cai, X. Zhang, X. Wang, and M. Li, "A Novel Learnable Dictionary Encoding Layer for End-to-End Language Identification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5189–5193.

[8] J. Chen, W. Cai, D. Cai, Z. Cai, H. Zhong, and M. Li, "End-to-end Language Identification using NetFV and NetVLAD," in *The 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2018.

[9] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A Discriminative Feature Learning Approach for Deep Face Recognition," in *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, 2016, pp. 499–515.

[10] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep Hypersphere Embedding for Face Recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 212–220.

[11] W. Cai, Z. Cai, W. Liu, X. Wang, and M. Li, "Insights into End-to-End Learning Scheme for Language Identification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5209–5213.

[12] W. Cai, J. Chen, and M. Li, "Analysis of Length Normalization in End-to-End Speaker Verification System," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018.

[13] D. Cai, X. He, K. Zhou, J. Han, and H. Bao, "Locality Sensitive Discriminant Analysis," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2007, pp. 1713–1726.

[14] D. Cai, W. Cai, Z. Ni, and M. Li, "Locality Sensitive Discriminant Analysis for Speaker Verification," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2016.

[15] B. Sun, J. Feng, and K. Saenko, "Return of Frustratingly Easy Domain Adaptation," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, 2016, pp. 2058–2065.

[16] M. J. Alam, G. Bhattacharya, and P. Kenny, "Speaker Verification in Mismatched Conditions with Frustratingly Easy Domain Adaptation," in *Odyssey 2018: The Speaker and Language Recognition Workshop*, 2018.

[17] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The Speakers in the Wild (SITW) Speaker Recognition Database," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016, pp. 818–822.

[18] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector Length Normalization in Speaker Recognition Systems," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2011, pp. 249–252.

[19] P. Matjka, O. Novotn, O. Plchot, L. Burget, M. D. Snchez, and J. ernock, "Analysis of Score Normalization in Multilingual Speaker Recognition," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017.

[20] N. Brümmer and E. De Villiers, "The BOSARIS Toolkit: Theory, Algorithms and Code for Surviving the New DCF," *arXiv preprint arXiv:1304.2865*, 2013.

[21] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A Large-Scale Speaker Identification Dataset," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 2616–2620.

[22] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep Speaker Recognition," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018.

[23] 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)*, 2017, pp. 5220–5224.

[24] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," *arXiv:1510.08484 [cs]*, 2015.