The SNU-HIL System Description for the SdSV Challenge 2020

Sung Hwan Mun, Woo Hyun Kang, Min Hyun Han, and Nam Soo Kim

Department of Electrical and Computer Engineering and the Institute of New Media and Communications, Seoul National University, Seoul, South Korea

{shmun, whkang, mhhan}@hi.snu.ac.kr, nkim@snu.ac.kr

Abstract

This paper describes the submission of Seoul National University Human Interface Lab (SNU-HIL) to the Short-duration Speaker Verification (SdSV) Challenge 2020. We focused on Task1 of the challenge, which was text-dependent speaker verification. The submitted our systems were composed of TDNN-based and ResNet-based front-end architectures with different pooling methods and objective functions. We also leveraged state-of-the-art automatic speech recognition (ASR) model for our proposed pooling strategy considering the lexical content and for estimating the posterior of phrase for score compensation. The performance of systems was verified through cosine similarity back-end and score normalization was applied for corresponding systems. On Task1's evaluation subset of the SdSV Challenge 2020, our single system achieved 0.1307 MinDCF and 3.18% EER. The finally submitted primary system was the fusion of different systems with score compensation and it obtained 0.0785 MinDCF and 2.23% EER on the challenge's evaluation subset. Besides, we reported the results of Task2 submitted to the challenge.

Index Terms: SdSV Challenge 2020, speaker verification.

1. Introduction

We present the submission of Seoul National University Human Interface Laboratory (SNU-HIL) to the Short-duration Speaker Verification (SdSV) Challenge 2020. The main purpose of this challenge is to evaluate new techniques for text-dependent and text-independent speaker verification in a short duration scenario [1]. The evaluation dataset used for the SdSV Challenge 2020 was derived from the multi-purpose DeepMine dataset [2, 3]. We mainly submitted our systems to Task 1 of the SdSV Challenge 2020, which was focused on text-dependent speaker verification (TD-SV).

In this work, our contribution is twofold. The first was the aggregation method using the frame-level estimated posteriors. We showed that this method is valid and effective in the text-dependent task through the results on the challenge's progress and evaluation subsets. The second was an effort to apply the techniques used in the text-independent task, which is optimized to separate only the speaker, to the text-dependent task. Instead of optimizing to separate both the speaker and the phrase at the same time, we firstly focused on discriminating each class well individually and then combined the two results later through the score compensation approach. For the speaker embedding network training, we used the popular techniques in the text-independent task. In terms of classifying the phrase, we leveraged the estimated frame-level posteriors of phrase using the state-of-the-art automatic speech recognition (ASR) model that outputs a probability distribution over characters per frame. Also, we verified that the combination of the results obtained from both processes significantly improve the performance, and the fusion of different systems we used produced the best performance in the SdSV Challenge 2020 task1. On top of that, we reported the results submitted to task 2.

The rest of this paper is organized as follows: Section 2 describes all components of our systems and Section 3 details the experimental conditions. Section 4 presents the results on the challenge's trial subsets we submitted. Finally, we conclude in section 5.

2. System components description

2.1. Front-end

In our systems, we used two types of front-end networks for extracting utterance-level speaker embedding: TDNN-based systems [4] and Thin ResNet34-based architectures [5].

2.1.1. TDNN-based architecture

The configuration of TDNN-based systems was based on the standard Kaldi recipe for the x-vector system [4, 6] and further, we made use of its modified structure shown as Table 1 for two purposes: leveraging character-level pooling strategy described in Section 2.2 and estimating the posteriors of phrase for score compensation described in Section 2.5. The input acoustic feature used in this architectures was log Mel-filterbank energies calculated from 20ms windows with a 10ms hop size and extracted by utilizing Librosa toolkit [7]. We selected 512 and 580 speaker embedding dimensions for statistics pooling and character-level pooling respectively. Our implementation and speaker embedding network training was based on Tensorflow toolkit [8].

2.1.2. ResNet-based architecture

The second front-end network was the ResNet-based system. We employed Thin ResNet34 architecture recently proposed by the authors in [5] (See Table 2). Compared to the original ResNet [9], Thin ResNet34 has only a quarter of channels in each residual block. We used 257-dimensional short-time Fourier transform (STFT) with 200-300 frames crop as input acoustic feature in this architecture and chose 256 dimensions for utterance-level speaker embedding. For implementation and training, we used Pytorch toolkit [10] and developed the systems based on the architectures in [11].

2.2. Pooling methods

In the TI-SV task, recently, numerous pooling mechanisms have been proposed such as a statistics pooling [4], a self-attentive pooling [12], a learnable dictionary encoding (LDE) pooling [13], a mutual information neural estimate (MINE) based pooling [14]. In this work, we employed various pooling methods proposed in the TI-SV task. Also, we propose a pooling strat-

Table 1: *TDNN-based front-end configuration for character-level pooling and score compensation.* $(d \times n)$ *indicates concatenation of n vectors, where the dimensionality of each vector is d. T: The number of segment frames, N: The number of speakers, M: The number of phrase types, CLP: Character-Level Pooling, LC: Locally-Connected, FC: Fully-Connected, BN: Batch Normalization*

Layer	Configuration for character-level pooling method			Configuration for score compensation method		
	TDNN	**Context**	**Output Size**	**TDNN**	**Context**	**Output Size**
Input	Log Mel-FBANK	-	$64 \times T$	Log Mel-FBANK	-	$64 \times T$
Frame1	512, stride 2, ReLU, BN	5, [t-2 : t+2]	$512 \times T$	1536, stride 2, ReLU, BN	5, [t-2 : t+2]	$1536 \times T$
Frame2	512, stride 1, ReLU, BN	3, [t-2, t, t+2]	$512 \times T$	512, stride 1, ReLU, BN	3, [t-2, t, t+2]	$512 \times T$
Frame3	512, stride 1, ReLU, BN	3, [t-3, t, t+3]	$512 \times T$	512, stride 1, ReLU, BN	3, [t-3, t, t+3]	$512 \times T$
Frame4	512, stride 1, ReLU, BN	1, [t]	512, stride 1	256, stride 1, ReLU, BN	1, [t]	$256 \times T$
Frame5	1536, stride 1, ReLU, BN	1, [t]	$1536 \times T$	256, stride 1, ReLU, BN	1, [t]	$256 \times T$
Pooling	CLP	T, [1 : T]	$(1536 \times 29) \times 1$	CLP	T, [1 : T]	$(256 \times 29) \times 1$
Segment1	LC *(speaker embedding)*	T, [1 : T]	$(20 \times 29) \times 1$	LC	T, [1 : T]	$(20 \times 29) \times 1$
Segment2	FC	T, [1 : T]	512×1	FC	T, [1 : T]	512×1
Softmax	FC	T, [1 : T]	$N \times 1$	FC *(posterior of phrase)*	T, [1 : T]	$M \times 1$

Table 2: *Thin ResNet34-based front-end configuration.*

Layer	Thin ResNet34	Output Size
Input	STFT	$257 \times T \times 1$
Conv1	$7\times7,\ 16,\ \text{stride } 2$	$129 \times T/2 \times 16$
	$3\times3,\ \text{max pooling},\ \text{stride } 2$	$65 \times T/4 \times 16$
Conv2	$\begin{bmatrix}3\times3,\ 16\\3\times3,\ 16\end{bmatrix}\times3,\ \text{stride } 1$	$65 \times T/4 \times 16$
Conv3	$\begin{bmatrix}3\times3,\ 32\\3\times3,\ 32\end{bmatrix}\times4,\ \text{stride } 2$	$33 \times T/8 \times 32$
Conv4	$\begin{bmatrix}3\times3,\ 64\\3\times3,\ 64\end{bmatrix}\times6,\ \text{stride } 2$	$17 \times T/16 \times 64$
Conv5	$\begin{bmatrix}3\times3,\ 128\\3\times3,\ 128\end{bmatrix}\times3,\ \text{stride } 2$	$9 \times T/32 \times 128$
FC	$9\times1,\ 256,\ \text{stride } 1$	$1 \times T/32 \times 256$

egy suitable for the text-dependent task by leveraging a frame-level probability distribution of each character estimated from the CTC-based ASR model. Pooling methods we used are as follows:

- Statistics Pooling [4]
- Self-Attentive Pooling [12]
- GhostVLAD Pooling [13]
- Character-Level Pooling

2.2.1. Character-level pooling

For extraction of utterance-level representations, The statistics pooling (SP) computes mean and standard deviation over the frame-level features. In the attentive pooling (SAP), the attention score is computed for each frame. In the meantime, the LDE pooling assumes that frame-level features are distributed in C clusters and it learns a dictionary with the centers of those clusters.

To consider the lexical context in TD-SV task, instead, we focused on the probability distribution of each character per frame-level feature. The probability of a character given a frame-level feature, i.e., the posterior, is denoted by:

$$\pi_{k,i} = P(\tilde{C} = c_k | \boldsymbol{h_i}) \tag{1}$$

where the set $\tilde{C} = \{c_k |\ c_k\ is\ k^{th} character,\ 1 \leq k \leq K\}$, and $\boldsymbol{h_i}$ is i^{th} frame-level feature with D_1 dimensions where $1 \leq i \leq T$. K indicates the number of symbols in the character set, and T is the number of segment frames. To estimate $\pi_{k,i}$, we leveraged decoder outputs of CTC-based end-to-end ASR model, termed $Jasper$, proposed [15]. Since this model was trained by using the Connectionist Temporal Classification (CTC) loss, our character set consisted of a total of 29 symbols including all alphabets (a-z), space, and the apostrophe symbol and the blank symbol used by the CTC loss. Then, the aggregation for character-level representation is as follows:

$$\boldsymbol{s_k} = \frac{\sum_{i=1}^{T} \pi_{k,i} \boldsymbol{h_i} + \tau}{\sum_{i=1}^{T} \pi_{k,i} + \tau} \tag{2}$$

$$\boldsymbol{s} = \left(\boldsymbol{s_1}^T \mid\ \ldots\ \mid \boldsymbol{s_K}^T\right)^T \tag{3}$$

where τ is a constant added to avoid divergence. All the character-level representations are concatenated as $\boldsymbol{s}$, and then it's passed through the locally-connected layer, which has K-part fully-connected layers for reducing dimensions and taking character-level affine transformation.

$$\boldsymbol{e_k} = f(\boldsymbol{W_k s_k} + \boldsymbol{b_k}) \tag{4}$$

$$\boldsymbol{e} = \left(\boldsymbol{e_1}^T \mid\ \ldots\ \mid \boldsymbol{e_K}^T\right)^T \tag{5}$$

Where W_k and $\boldsymbol{b_k}$ indicate trainable parameters with $D_2 \times D_1$ and D_2 dimensions respectively and $f(\cdot)$ means a non-linear activation function. Finally, we can obtain an utterance-level embedding $\boldsymbol{e}$ (See Table 1).

2.3. Objective functions

In our work, we used various objective functions. The first one is the classification loss based on softmax, which is formulated by a multi-class cross-entropy loss. Over the last few years, the different variants have been proposed to overcome the limitations of the standard softmax. We employed some of them in our systems. Secondly, we used end-to-end based losses, which directly optimize the distance metrics such as Euclidean or Cosine distance and exactly imitate the test scenario during training. The objective functions used in our systems are followed as:

- Standard Softmax
- Additive Margin Softmax (AM-Softmax) [16, 17]
- Additive Angular Margin Softmax (AAM-Softmax) [18]
- Angular Prototypical Loss (A-Prototypical) [11]
- Generalized End-to-End Loss (GE2E) [19]

2.4. Back-end

In the back-end module, we only used cosine similarity as a scoring method between the two speaker embeddings. Neither LDA nor WCCN was applied, and the results of PLDA were not reported in this works because we did not observe better performance.

2.5. Score normalization

To minimize the domain mismatch (e.g. languages, recording environments, etc.) between the training and the evaluation set and to normalize the distribution of scores during fusion between different models, we used the score normalization technique, which was the Adaptive Symmetric Score Normalization (AS-Norm) [20]. We selected speaker-phrase dependent models in DeepMine Task1 Train Partition (i.e. in-domain training data) as cohort set and used the most similar top 300 scoring files to calculate normalization variables of enrollment and test sets respectively.

2.6. Score compensation

Instead of TI-SV's approaches learned via maximizing the between-class (inter-speaker) difference and minimizing the within-class (intra-speaker) variation at the same time, we firstly focused on separating each class well and then fused the two results later by using the score compensation approach. First, we define the posteriors of the phrase as follows:

$$\boldsymbol{u_X} = \big(P(\tilde{U} = u_1|\boldsymbol{X}),\ \ldots\ ,\ P(\tilde{U} = u_M|\boldsymbol{X})\big)^T \tag{6}$$

$$\sum_{j=1}^{M} p(\tilde{U} = u_j|\boldsymbol{X}) = 1 \tag{7}$$

where the set $\tilde{U} = \{u_j|\ u_j\ is\ j^{th} phrase,\ 1 \leq j \leq M\}$, M is the number of phrase types in TD-SV's dataset, X is an acoustic feature such as MFCCs. We estimate the posterior $p(\tilde{U} = u_j|X)$ using softmax layers of TDNN-based network. The architecture of this network is identical to that of TDNN-based front-end described as Section 2.1, but composed of smaller size layers to prevent overfitting: the five frame-level TDNN layers with $\{1536, 512, 512, 256, 256\}$ size, the character-level pooling layer with (256×29) size, and a softmax output layer with M size (See Table 1). For training the network, we used the DeepMine Task1 Train Partition which includes 10 types of phrases. Finally, we compute the compensation factor and the total score between X and Y as follows:

$$c^{phr}_{X,Y} = \boldsymbol{u_X}^T\, \boldsymbol{u_Y} \tag{8}$$

$$s_{X,Y} = \tilde{s}^{spk}_{X,Y} + \alpha c^{phr}_{X,Y} \tag{9}$$

where $c^{phr}_{X,Y}$ is compensation factor, $\tilde{s}^{spk}_{X,Y}$ is the normalized (AS-Norm) score between embeddings of X and Y, α is a scale factor, and $s_{X,Y}$ is the total score.

3. Experimental conditions

3.1. Training condition

According to the fixed training condition of the challenge, we used the designated datasets for training our systems and utilized RSR2015 dataset [21] as a validation set for monitoring. The training set for each system was the combination of different datasets and each training dataset is described as follows.

3.1.1. DeepMine (Task 1 Train Partition)

This is the main dataset for Task 1, i.e., in-domain data, of the SdSV Challenge. It contains 101,063 utterances from 963 speakers, which has five Persian phrases as well as five English. We used it for training (1) speaker embedding networks and (2) the posterior estimator network, and also as (3) cohort set to calculate parameters of AS-Norm.

3.1.2. VoxCeleb1 & 2

The training data includes VoxCeleb1 [22] and VoxCeleb2 [23]. We only used the development sets of both datasets, which consist of 148,642 and 1,092,009 utterances from 1,211 and 5,994 speakers respectively. In our systems, they were used to train the speaker embedding networks.

3.1.3. LibriSpeech

To train the CTC-based ASR model, namely $Jasper$, which was utilized in character-level pooling and for estimating the posterior of phrase, we used the train-clean/other sets of LibriSpeech corpus [24], which comprises 281,241 utterances from 2,338 speakers. Additionally, in some of our systems, we employed them for training speaker embedding networks.

3.1.4. DeepMine (Task 2 Train Partition)

This is the main dataset for Task 2 of the SdSV Challenge. It includes 85,764 utterances from 588 speakers, which has text-independent Persian utterances. We used it for training the speaker embedding networks in case of only Task 2.

3.2. Trial condition

3.2.1. Task 1: Text-dependent speaker verification

According to the trial condition of the challenge, the enrollment was accomplished using three utterances of a specific phrase for each model and among four types of trials in the TD-SV task, only Target-Correct, where the target speaker utters the correct pass-phrase, was considered as target and the rest was an imposter. The whole set of trials was divided into two subsets: a progress subset (30%), and an evaluation subset (70%). The progress subset was used to monitor progress on the leaderboard, while the evaluation subset was used for the official results.

3.2.2. Task 2: Text-independent speaker verification

In the trial set of Task 2, The enrollment models were obtained using one to several utterances assigned from the SdSV challenge. Similar to Task 1, The test set included two subsets: a progress subset (30%), and an evaluation subset (70%) for the same purpose as task 1.

Table 3: ***Task 1.*** *Results on the Trial Subsets for the SdSV Challenge 2020* ***without AS-Norm & Score Compensation****. TDT: Text-Dependent Training, CLP: Character-Level Pooling, SP: Statistics Pooling, GVP: GhostVLAD Pooling, SAP: Self-Attentive Pooling. Deep1: DeepMine Task1, Vox1: VoxCeleb1, Vox2: VoxCeleb2, Libri: LibriSpeech.*

#	Front-End	Objectives	Pooling	Training Dataset	Progress subset		Evaluation subset	
					MinDCF	EER[%]	MinDCF	EER[%]
1				Deep1	0.3755	9.19	0.3775	9.18
2	TDNN	**Softmax**	**CLP**	**Deep1 / Vox1**	0.3571	**8.45**	0.3585	**8.48**
3				Deep1 / Vox1 / Vox2	0.4044	8.97	0.4066	9.00
4	**TDNN**	**Softmax (TDT)**	**CLP**	**Deep1**	**0.3547**	8.82	**0.3554**	8.88
5				Deep1	0.8679	17.18	0.8688	17.25
6	TDNN	Softmax	SP	Deep1 / Vox1	0.7636	14.37	0.7641	14.45
7				Deep1 / Vox1 / Vox2	0.6511	12.71	0.6539	12.77
8	ResNet34	Softmax	GVP	Vox2	0.8891	14.84	0.8897	14.87
9	ResNet34	AAM-Softmax	SAP	Deep1 / Vox1 / Vox2	0.9030	16.09	0.9021	16.12
10				Deep1 / Vox1 / Vox2 / Libri	0.9157	16.39	0.9159	16.45
11	ResNet34	AM-Softmax	SAP	Deep1 / Vox1 / Vox2	0.8944	15.76	0.8931	15.84
12				Deep1 / Vox1 / Vox2 / Libri	0.9195	16.42	0.9181	16.47
13	ResNet34	A-Prototypical	SAP	Vox2	0.7957	13.35	0.7973	13.33
14				Deep1 / Vox1 / Vox2	0.8659	15.92	0.8652	15.98
15	ResNet34	GE2E	SAP	Deep1 / Vox1 / Vox2	0.9226	16.39	0.9212	16.45
16	x-vector baseline *(provided by Organizers in Task 1* [1]*)*				0.5290	9.05	0.5287	9.05

Table 4: ***Task 1.*** *Results on the Trial Subsets for the SdSV Challenge 2020* ***with AS-Norm & Score Compensation and The Fusion.*** *The Fusion is equal-weighted average.*

#	Front-End	Objectives	Pooling	Training Dataset	Progress subset		Evaluation subset	
					MinDCF	EER[%]	MinDCF	EER[%]
1				Deep1	0.2164	5.79	0.2185	5.82
2	TDNN	Softmax	CLP	Deep1 / Vox1	0.1845	4.72	0.1856	4.80
3				Deep1 / Vox1 / Vox2	0.1892	4.91	0.1918	4.98
4	TDNN	Softmax (TDT)	CLP	Deep1	0.2327	5.88	0.2333	5.98
5				Deep1	0.2540	7.35	0.2554	7.42
6	**TDNN**	**Softmax**	**SP**	Deep1 / Vox1	0.2069	5.54	0.2085	5.63
7[†]				**Deep1 / Vox1 / Vox2**	**0.1730**	**4.49**	**0.1753**	**4.55**
8	ResNet34	Softmax	GVP	Vox2	0.1993	4.59	0.2017	4.65
9	ResNet34	AAM-Softmax	SAP	Deep1 / Vox1 / Vox2	0.1327	3.15	0.1332	3.21
10				Deep1 / Vox1 / Vox2 / Libri	0.1321	3.30	0.1325	3.33
11	**ResNet34**	**AM-Softmax**	**SAP**	**Deep1 / Vox1 / Vox2**	**0.1299**	**3.13**	**0.1307**	**3.18**
12				Deep1 / Vox1 / Vox2 / Libri	0.1387	3.53	0.1395	3.58
13	ResNet34	A-Prototypical	SAP	Vox2	0.1762	3.99	0.1769	3.96
14				Deep1 / Vox1 / Vox2	0.1647	3.83	0.1654	3.85
15	ResNet34	GE2E	SAP	Deep1 / Vox1 / Vox2	0.1768	4.08	0.1778	4.07
16	x-vector baseline *(provided by Organizers in Task 1* [1]*)*				0.5290	9.05	0.5287	9.05
17	i-vector/HMM baseline *(provided by Organizers in Task 1* [1]*)*				0.1472	3.47	0.1464	3.49
18	Fusion of TDNNs [1-7]				0.1242	3.50	0.1257	3.55
19	Fusion of ResNet34s [8-14]				0.0940	2.40	0.0942	2.42
20[‡]	**Fusion of all systems [1-14]**				**0.0771**	**2.18**	**0.0785**	**2.23**

[†] The single system we submitted.
[‡] The primary system we submitted.

Table 5: ***Task 2.*** *Results on the Trial Subsets for the SdSV Challenge 2020. Deep2: DeepMine Task2.*

#	Front-End	Objectives	Pooling	Training Dataset	Progress subset		Evaluation subset	
					MinDCF	EER[%]	MinDCF	EER[%]
21	TDNN	Softmax	SP	Deep2 / Vox1 / Vox2	0.3015	6.02	0.3005	6.03
22†	**ResNet34**	**Softmax**	**GVP**	**Vox2**	**0.2451**	**4.88**	**0.2438**	**4.88**
23	x-vector baseline *(provided by Organizers in Task 2* [1]*)*				0.4319	10.67	0.4324	10.67
24‡	**Fusion of all systems [21-22] (weighted average***)				**0.2078**	**3.96**	**0.2073**	**3.95**

† The single system we submitted.
‡ The primary system we submitted.
* The weights were hand-picked.

4. Analysis

We analyzed two experimental scenarios in only Task 1. In Task 1, firstly we verified the feasibility and effectiveness of the character-level pooling strategy for TD-SV task through the results of the progress and evaluation subsets (Table 3). In the second experiment, we applied AS-Norm and score compensation to all the systems we used in the first experiment, to further boost the performance. Also, we fused different systems and confirmed the best primary system and single system on the progress and evaluation subsets in terms of MinDCF, which was the main metric for the challenge (Table 4). Additionally, we reported the results submitted to task 2 (Table 5), and the analysis of task 2 was omitted. No preprocessing such as data augmentation or VAD were applied to the training and trial data for both tasks.

4.1. Analysis of character-level pooling method in Task 1

Each subsystem (1-15) was a composite of different front-ends, pooling techniques, objectives, and training datasets described in Section 2 and 3.1. Among them, system (5) utilized text-dependent training (TDT), which was jointly trained by combined classes of speaker and phrase (i.e., speakers $\times$ phrases classes). In systems of (5-16), phrase information was not considered, since they were trained for TI-SV. For the reasons stated in Section 2.2, the character-level pooling methods (systems 1-4) showed improved performances compared with other systems in terms of 0.3554 MinDCF (system 4) and 8.48 EER (system 2) on the challenge's evaluation subsets.

4.2. Results using AS-Norm & compensation in Task 1

We used AS-Norm and score compensation described in Sections 2.6 and 2.7 respectively, for improvement of performances. As you can see in Table 4, the performance of all systems was improved significantly. In particular, the performances of systems that didn't consider the lexical context (5-15) increased greatly, compared to the character-level pooling systems (1-4), which showed minor improvement. The best performance of a single system was 0.1307 MinDCF and 3.18 EER on the evaluation subsets (system 11). Finally, we fused different models, which were TDNN-based (18), ResNet-based (19), and all systems (20). Overall, the performance of ResNet-based systems outperformed TDNN-based networks in the case of both single systems and fusions, and the best primary system was the fusion of all systems. It obtained 0.0785 MinDCF and 2.23% EER on the evaluation subset.

5. Conclusions

the submission of Seoul National University Human Interface Lab (SNU-HIL) to the SdSV Challenge 2020. We propose a new pooling and score compensation methods that leverage a CTC-based end-to-end ASR model for taking the lexical content into account. Our systems contained two front-end architectures and acoustic features, and various pooling methods including our proposal, and different objective functions. Experiments show that the usage of the proposed character-level pooling and score compensation methods significantly enhances text-dependent speaker verification performance. In Task 1, finally the best performance of the primary system was obtained through the fusion of all the experimented systems, which showed 0.0785% MinDCF and 2.23% EER on the challenge's evaluation subset.

6. Acknowledgements

This research was supported and funded by the Korean National Police Agency. [Project Name: Real-time speaker recognition via voiceprint analysis / Project Number: PA-J000001-2017-101].

7. References

[1] H. Zeinali, K. A. Lee, J. Alam, and L. Burget, "Short-duration Speaker Verification (SdSV) challenge 2020: The challenge evaluation plan," *arXiv preprint arXiv:1912.06311*, 2019.

[2] 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 *Proc. The Speaker and Language Recognition Workshop (Speaker Odyssey)*, 2018, pp. 386–392.

[3] H. Zeinali, L. Burget, J. Černockỳ *et al.*, "A multi purpose and large scale speech corpus in persian and english for speaker and speech recognition: The DeepMine database," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019.

[4] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[5] J. S. Chung, J. Huh, and S. Mun, "Delving into VoxCeleb: Environment invariant speaker recognition," in *Proc. The Speaker and Language Recognition Workshop (Speaker Odyssey)*, 2020, to be published.

[6] 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," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011.

[7] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proc. The 14th Python in Science Conference (SciPy)*, vol. 8, 2015.

[8] 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 *Proc. Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[10] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. Conference and Workshop on Neural Information Processing Systems (NIPS Workshop)*, 2017.

[11] 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," *arXiv preprint arXiv:2003.11982*, 2020.

[12] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 3573–3577.

[13] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5791–5795.

[14] M. H. Han, W. H. Kang, S. H. Mun, and N. S. Kim, "Information preservation pooling for speaker embedding," in *Proc. The Speaker and Language Recognition Workshop (Speaker Odyssey)*, 2020, to be published.

[15] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, "Jasper: An end-to-end convolutional neural acoustic model," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 71–75.

[16] 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.

[17] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5265–5274.

[18] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4690–4699.

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

[20] P. Matejka, O. Novotnỳ, O. Plchot, L. Burget, M. D. Sánchez, and J. Cernockỳ, "Analysis of score normalization in multilingual speaker recognition." in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 1567–1571.

[21] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Communication*, vol. 60, pp. 56–77, 2014.

[22] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 2616–2620.

[23] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 1086—1090.

[24] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.