*INTERSPEECH 2020*
October 25–29, 2020, Shanghai, China

# Robust Text-Dependent Speaker Verification via Character-Level Information Preservation 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 our submission to Task 1 of the Short-duration Speaker Verification (SdSV) challenge 2020. Task 1 is a text-dependent speaker verification task, where both the speaker and phrase are required to be verified. The submitted systems were composed of TDNN-based and ResNet-based front-end architectures, in which the frame-level features were aggregated with various pooling methods (e.g., statistical, self-attentive, ghostVLAD pooling). Although the conventional pooling methods provide embeddings with a sufficient amount of speaker-dependent information, our experiments show that these embeddings often lack phrase-dependent information. To mitigate this problem, we propose a new pooling and score compensation methods that leverage a CTC-based automatic speech recognition (ASR) model for taking the lexical content into account. Both methods showed improvement over the conventional techniques, and the best performance was achieved by fusing all the experimented systems, which showed 0.0785% MinDCF and 2.23% EER on the challenge's evaluation subset.
**Index Terms**: SdSV Challenge 2020, speaker verification.

## 1. Introduction

This paper presents our submission to Task 1 of the Short-duration Speaker Verification (SdSV) Challenge 2020. The main purpose of this challenge is to evaluate new techniques for 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 submitted the systems to Task 1 of the SdSV Challenge 2020, which was focused on text-dependent speaker verification (TD-SV).

During the past decade, there has been significant improvement in the field of text-independent speaker verification (TI-SV), which is mainly attributed to the development of deep neural network (DNN) based speaker embedding. These mechanisms have been developed through a variety of methods such as deeper architectures [4, 5, 6, 7], pooling strategies [6, 8, 9, 10], and different objective functions [11, 12, 13, 14, 15, 16]. However, these techniques are optimized to discriminate only the speaker and may be ineffective in the TD-SV task in which the lexical context, as well as the speaker, is considered [17].

On the other hand, there have also been various efforts to boost performance in the TD-SV. In [18], Larcher et al. used an HMM-based system named HiLAM to model each speaker and each senone state. H. Zeinali et al. [17] proposed a straightforward HMM-based extension of the i-vector approach [19], which allows i-vectors to contain sufficient text-dependent information. In [20], Y. Lei et al. used DNN to estimate the posteriors of the frames for calculation of sufficient statistics and E. Variani et al. [21] extracted frame-level representations termed d-vector through a hidden layer of DNN for the TD-SV task. In [22], Matejka et al. employed bottle-neck DNN features concatenated to other acoustic features to improve the performance, and Zhang et al. [23] proposed an attention aggregation-based end-to-end TD-SV system which takes the speaker and phonetic information into account.

In this paper, we focus on preserving the character-level information. Overall, our contributions are as follows:

- **Character-level pooling:** We introduce the aggregation method using the estimated frame-level posterior obtained from an automatic speech recognition (ASR) model. Experiments show that this method is valid and effective in the TD-SV through the results on the challenge's progress and evaluation subsets.
- **Score compensation:** We propose the score compensation method where the probability of pass-phrase is estimated. Our experiments show that the usage of the proposed score compensation significantly enhances performance in the TD-SV even if the embedding network is trained only to classify the speaker.

Furthermore, the fusion of different systems we employed produces the best performance on the challenge's trial subsets.

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

## 2. System components description

### 2.1. Front-end

In our systems, we used two types of front-end networks: TDNN-based [6] and ResNet-based [5] architectures.

**TDNN-based architecture**. The configuration of TDNN-based systems is shown in Table 1. The usage for each system is described in Section 2.2 and 2.6. The input acoustic feature used in this architectures was log Mel-filterbank energies calculated from 20ms windows with a 10ms hop size and extracted via the Librosa toolkit [24]. We selected 512 and 580 speaker embedding dimensions for statistics pooling and character-level pooling respectively. Our implementation and speaker embedding network training was done using Tensorflow toolkit [25].

**ResNet-based architecture**. We used Thin ResNet34 architecture recently proposed in [5] (Table 2). Compared to the original ResNet [4], it has only a quarter of channels in each residual block. In this architecture, we used 257-dimensional short-time Fourier transform (STFT) with 200-300 frames crop as input acoustic feature and chose 512 dimensions for embedding. For implementation, we used Pytorch toolkit [26] and developed the systems based on the architectures in [15].



http://dx.doi.org/10.21437/Interspeech.2020-2183

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*] | $512 \times T$ | 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*] | $512 \times 1$ | FC | *T*, [1 : *T*] | $512 \times 1$ |
| 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. All convolutional layers have 1 zero-padding.*

| **Layer** | **Thin ResNet34** | **Output Size** |
|---|---|---|
| Input | STFT | $257 \times T \times 1$ |
| Conv1 | $7\times7,\ 16,$ stride 2<br>$3\times3,$ max pooling, stride 2 | $129 \times T/2 \times 16$<br>$65 \times T/4 \times 16$ |
| Conv2 | $\begin{bmatrix} 3\times3,\ 16 \\ 3\times3,\ 16 \end{bmatrix} \times 3$, stride 1 | $65 \times T/4 \times 16$ |
| Conv3 | $\begin{bmatrix} 3\times3,\ 32 \\ 3\times3,\ 32 \end{bmatrix} \times 4$, stride 2 | $33 \times T/8 \times 32$ |
| Conv4 | $\begin{bmatrix} 3\times3,\ 64 \\ 3\times3,\ 64 \end{bmatrix} \times 6$, stride 2 | $17 \times T/16 \times 64$ |
| Conv5 | $\begin{bmatrix} 3\times3,\ 128 \\ 3\times3,\ 128 \end{bmatrix} \times 3$, stride 2 | $9 \times T/32 \times 128$ |
| FC | $9\times1,\ 512,$ stride 1 | $1 \times T/32 \times 512$ |

### 2.2. Pooling methods

In the TI-SV task, various pooling mechanisms have been proposed such as statistics pooling [6], self-attentive pooling [8], learnable dictionary encoding (LDE) pooling [9], mutual information neural estimate (MINE) based pooling [10]. In this work, we employed a variety of pooling methods proposed in the TI-SV task. On top of that, we propose a pooling strategy suitable for the TD-SV, i.e., character-level pooling, which leverages a frame-level probability distribution of each character estimated from the end-to-end based ASR model. The pooling methods we used are as follows:

- Statistics Pooling (SP) [6]
- Self-Attentive Pooling (SAP) [8]
- GhostVLAD Pooling (GVP) [9]
- Character-Level Pooling (CLP)

**Character-level pooling.** To extract the utterance-level representation appropriate for the TD-SV, we exploit the character posterior probabilities of each frame-level feature. The probability of a character given a frame-level feature, i.e., the posterior, is denoted by:

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

where the set $C = \{c_k |\ c_k\ is\ k^{th} character,\ 1 \leq k \leq K\}$, and $\boldsymbol{h_i}$ is the $i^{th}$ 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 the decoder outputs of the end-to-end based ASR model, termed $Jasper$, proposed by [27]. 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), the space symbol, the apostrophe symbol and the blank symbol used by the CTC loss. Then, the aggregation for character-level representation is as follows:

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

$$\boldsymbol{v} = \left(\boldsymbol{v_1}^T \mid \ \ldots \ \mid \boldsymbol{v_K}^T\right)^T \tag{3}$$

where $\tau$ is a constant added to avoid divergence. All the character-level representations are concatenated as $\boldsymbol{v}$, 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.

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

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

Where $\boldsymbol{W_k}$ 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 made use of various objective functions conventionally used in speaker embedding training. Some variants of softmax-based classification loss were employed in our systems. Also, we used end-to-end based losses which directly optimize distance metrics such as Euclidean or Cosine distance. The objective functions used in the systems are as follows:

- Standard Softmax
- Additive Margin Softmax (AM-Softmax) [12, 13]
- Additive Angular Margin Softmax (AAM-Softmax) [14]
- Angular Prototypical Loss (A-Prototypical) [15]
- Generalized End-to-End Loss (GE2E) [16]

### 2.4. Back-end

In the back-end module, we only used cosine similarity as a scoring method between the two speaker embeddings. No Linear Discriminant Analysis (LDA), Within-Class Covariance Normalization (WCCN), and Probabilistic Linear Discriminant Analysis (PLDA) was applied in this work.

Table 3: *Results on the Trial Subsets for the SdSV Challenge 2020* ***without AS-Norm & Score Compensation****. TDT: Text-Dependent Training, Deep: DeepMine, Vox1: VoxCeleb1, Vox2: VoxCeleb2, Libri: LibriSpeech.*

| # | Front-End | Objectives | Pooling | Training Dataset | Progress subset | | Evaluation subset | |
|---|---|---|---|---|---|---|---|---|
| | | | | | **MinDCF** | **EER[%]** | **MinDCF** | **EER[%]** |
| 1 | | | | Deep | 0.3755 | 9.19 | 0.3775 | 9.18 |
| **2** | **TDNN** | **Softmax** | **CLP** | **Deep / Vox1** | 0.3571 | **8.45** | 0.3585 | **8.48** |
| 3 | | | | Deep / Vox1 / Vox2 | 0.4044 | 8.97 | 0.4066 | 9.00 |
| **4** | **TDNN** | **Softmax (TDT)** | **CLP** | **Deep** | **0.3547** | 8.82 | **0.3554** | 8.88 |
| 5 | | | | Deep | 0.8679 | 17.18 | 0.8688 | 17.25 |
| 6 | TDNN | Softmax | SP | Deep / Vox1 | 0.7636 | 14.37 | 0.7641 | 14.45 |
| 7 | | | | Deep / 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 | Deep / Vox1 / Vox2 | 0.9030 | 16.09 | 0.9021 | 16.12 |
| 10 | | | | Deep / Vox1 / Vox2 / Libri | 0.9157 | 16.39 | 0.9159 | 16.45 |
| 11 | ResNet34 | AM-Softmax | SAP | Deep / Vox1 / Vox2 | 0.8944 | 15.76 | 0.8931 | 15.84 |
| 12 | | | | Deep / 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 | | | | Deep / Vox1 / Vox2 | 0.8659 | 15.92 | 0.8652 | 15.98 |
| 15 | ResNet34 | GE2E | SAP | Deep / Vox1 / Vox2 | 0.9226 | 16.39 | 0.9212 | 16.45 |
| 16 | x-vector baseline *(provided by SdSV)* | | | | 0.5290 | 9.05 | 0.5287 | 9.05 |

### 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 Adaptive Symmetric Score Normalization (AS-Norm) [28]. 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

Since the TI-SV approaches focus on minimizing the within-speaker variability, the embedding vectors may lack crucial information on the lexical content. Thus such embedding vectors are improper for the TD-SV experiments. To complement the lost contextual discriminability, we introduce a score compensation method. Firstly, we define the posterior of the phrase as follows:

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

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

where the set $U = \{u_j|\ u_j\ is\ j^{th} phrase,\ 1 \leq j \leq M\}$, $M$ is the number of phrase types in the TD-SV's dataset, $X$ is an acoustic feature such as MFCC. We estimate the posterior $p(U = u_j|X)$ using softmax layers of TDNN-based network. The architecture of this network is identical to the configuration of TDNN-based architecture for CLP but composed of smaller size layers to prevent overfitting (See Table 1). For training the network, we used 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:

$$s^{phr}_{X,Y} = \boldsymbol{u_X}^T \boldsymbol{u_Y} \quad (8)$$

$$s_{X,Y} = \tilde{s}^{spk}_{X,Y} + \alpha s^{phr}_{X,Y} \quad (9)$$

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

## 3. Experimental conditions and Analysis

### 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 [18] 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.

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

**VoxCeleb1 & 2**. We used the development sets of VoxCeleb1 [29] and VoxCeleb2 [30], 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.

**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 [31], which comprise 281,241 utterances from 2,338 speakers. Additionally, in some systems, we employed them for training speaker embedding networks.

### 3.2. Trial condition

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 trial set was divided into two subsets: a progress

Table 4: *Results on the Trial Subsets* ***with AS-Norm & Score Compensation and the Fusions****. †: Fusion is equal-weighted sum.*

| # | Front-End | Objectives | Pooling | Training Dataset | Progress subset | | Evaluation subset | |
|---|---|---|---|---|---|---|---|---|
| | | | | | **MinDCF** | **EER[%]** | **MinDCF** | **EER[%]** |
| 1 | TDNN | Softmax | CLP | Deep | 0.2164 | 5.79 | 0.2185 | 5.82 |
| 2 | | | | Deep / Vox1 | 0.1845 | 4.72 | 0.1856 | 4.80 |
| 3 | | | | Deep / Vox1 / Vox2 | 0.1892 | 4.91 | 0.1918 | 4.98 |
| 4 | TDNN | Softmax (TDT) | CLP | Deep | 0.2327 | 5.88 | 0.2333 | 5.98 |
| 5 | TDNN | Softmax | SP | Deep | 0.2540 | 7.35 | 0.2554 | 7.42 |
| 6 | | | | Deep / Vox1 | 0.2069 | 5.54 | 0.2085 | 5.63 |
| 7 | | | | Deep / 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 | Deep / Vox1 / Vox2 | 0.1327 | 3.15 | 0.1332 | 3.21 |
| 10 | | | | Deep / Vox1 / Vox2 / Libri | 0.1321 | 3.30 | 0.1325 | 3.33 |
| **11** | **ResNet34** | **AM-Softmax** | **SAP** | **Deep / Vox1 / Vox2** | **0.1299** | **3.13** | **0.1307** | **3.18** |
| 12 | | | | Deep / 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 | | | | Deep / Vox1 / Vox2 | 0.1647 | 3.83 | 0.1654 | 3.85 |
| 15 | ResNet34 | GE2E | SAP | Deep / Vox1 / Vox2 | 0.1768 | 4.08 | 0.1778 | 4.07 |
| 16 | x-vector baseline *(provided by SdSV)* | | | | 0.5290 | 9.05 | 0.5287 | 9.05 |
| 17 | i-vector/HMM baseline *(provided by SdSV)* | | | | 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** |

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.

### 3.3. Analysis

We analyzed two experimental scenarios. First, we verified the feasibility and effectiveness of the character-level pooling strategy for the TD-SV task through the results on 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). No preprocessing such as data augmentation or VAD was applied to the training and trial data.

**Analysis of character-level pooling strategy (Table 3)**. Each subsystem (1-15) was composed 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 and 8.48% EER on the challenge's evaluation subsets (See Table 3).

**Results using AS-Norm & score compensation (Table 4)**. 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. From these results, we could interpret that the better the speaker is distinguished, the higher the performance can be achieved, when score compensation was applied. Finally, we performed the fusion by computing the equal-weighted sum of the scores of different systems, which were TDNN-based (18), ResNet-based (19), and all systems (20). Overall, the performance of ResNet-based systems outperformed TDNN-based systems 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.

## 4. Conclusions

In this paper, we described our submission to Task 1 of 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. 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.

## 5. Acknowledgements

This work was supported by the research fund of Signal Intelligence Research Center supervised by Defense Acquisition Program Administration and Agency for Defense Development of Korea.

# 6. 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, Tech. Rep., 2020.

[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] 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.

[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.

[6] 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.

[7] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5796–5800.

[8] 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.

[9] 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.

[10] 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.

[11] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 212–220.

[12] 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.

[13] 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.

[14] 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.

[15] 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 *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2020.

[16] 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.

[17] H. Zeinali, H. Sameti, and L. Burget, "HMM-based phrase-independent i-vector extractor for text-dependent speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 7, pp. 1421–1435, 2017.

[18] 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.

[19] 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, 2010.

[20] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1695–1699.

[21] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4052–4056.

[22] P. Matějka, O. Glembek, O. Novotnỳ, O. Plchot, F. Grézl, L. Burget, and J. H. Cernockỳ, "Analysis of DNN approaches to speaker identification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5100–5104.

[23] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 1487–1491.

[24] 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.

[25] 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.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Conference on Neural Information Processing Systems (NeurIPS)*, 2019, pp. 8024–8035.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.