# DEEP SPEAKER REPRESENTATION USING ORTHOGONAL DECOMPOSITION AND RECOMBINATION FOR SPEAKER VERIFICATION

*Insoo Kim* *Kyuhong Kim* *Jiwhan Kim* *Changkyu Choi*

Samsung Advanced Institute of Technology (SAIT), South Korea

## ABSTRACT

Speech signal contains intrinsic and extrinsic variations such as accent, emotion, dialect, phoneme, speaking manner, noise, music, and reverberation. Some of these variations are unnecessary and are unspecified factors of variation. These factors lead to increased variability in speaker representation. In this paper, we assume that unspecified factors of variation exist in speaker representations, and we attempt to minimize variability in speaker representation. The key idea is that a primal speaker representation can be decomposed into orthogonal vectors and these vectors are recombined by using deep neural networks (DNN) to reduce speaker representation variability, yielding performance improvement for speaker verification (SV). The experimental results show that our proposed approach produces a relative equal error rate (EER) reduction of 47.1% compared to the use of the same convolutional neural network (CNN) architecture on the VoxCeleb dataset. Furthermore, our proposed method provides significant improvement for short utterances.

***Index Terms***— speaker verification, speaker embedding, orthogonal vector pooling, deep learning, CNN

## 1. INTRODUCTION

Speaker verification (SV) is a technique to verify whether a speaker is enrolled for a given utterance. Recently, speaker verification has been used as a voice wake-up and authentication system. In particular, the high reliability should be ensured in authentication system. At Samsung electronics, we are interested in text-dependent speaker verification with the keyword "Hi Bixby" and a user-defined keyword which is used in the enrollment and evaluation phases. To improve the performance, we should investigate short utterances and phonetic variability to accommodate user-defined keywords.

Gaussian mixture model with universal background model (GMM-UBM) is a method for modeling a specific speaker model using the *maximum a posteriori* (MAP) adaptation [1]. Gaussian mixture model with support vector machine (GMM-SVM) technique was also used to model speaker supervectors [2]. Most speaker verification systems rely on the i-vector framework which generates a variation-independent vector from the total variability [3]. The i-vector framework shows outstanding performance in both text-dependent and text-independent speaker verification. However, the i-vector method suffers from performance degradation in short utterances due to the utterance variability. To overcome this limitation, many SV systems were implemented with pre-processing methods that reduces intra-speaker variability, such as nuisance attribute projection (NAP) [4], and within-class covariance normalization (WCCN) [5].

In recent years, deep neural networks (DNNs) based speaker representation were proposed to improve performance in short utterances. The text-dependent speaker representation method known as the d-vector was proposed [6]. They used four fully-connected layers (FCNs) and a softmax loss to maximize inter-speaker variability and minimize intra-speaker variability in speaker representations. A more reliable speaker representation with attention mechanisms was also proposed [7]. The CNN-based speaker representations were also investigated for many years [8–10]. A joint learning method is an approach that fuses an i-vector and deep embedding to take advantages of both methods [11]. The other method, known as the x-vector, provides better performance than the i-vector method in text-independent SV [12]. The x-vector is a statistically-based speaker representation method that uses statistical pooling. These DNN-based methods provide remarkable performance in text-independent SV. Even though previous works reported performance improvements, there has been little discussion of the variations in speaker representations.

In this paper, we assume that unspecified factors of variation exist in a speaker representation, and we attempt to reduce these factors. The main idea is that a speaker representation is set to a weighted sum of latent vectors, thus we can reduce unspecified variability, by learning the latent vectors and weights from the final feature maps. The better performance can be achieved if the state-of-the-art CNN architectures are employed, and the further performance improvement can be fulfilled when we apply our method to CNN architectures.

The remainder of this paper is organized as follows. Section 2 presents an overview of related works. In Section 3, the concept of orthogonal vector pooling and a recombination network are described. Experimental results are presented in Section 4. The paper is concluded in Section 5.

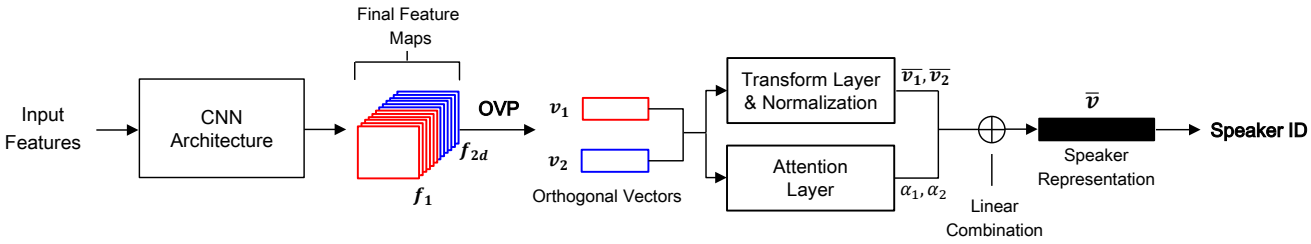



**Fig. 1**. Network architecture overview: orthogonal vector pooling and recombination network

## 2. RELATED WORK

### 2.1. Feature Representation in Learning Method

Discriminative speaker representations can be generated by modifying learning mechanisms. The CNN architectures were successfully applied to speaker verification [8–10]. Recently, curriculum learning was presented to improve both text-dependent and text-independent speaker verification performance [13]. Progressive learning was introduced to remove noise components [14]. This paper suggests a method for removing noise from raw data; however, it can be also used to extract feature representations. To cope with the domain mismatch problem, domain adversarial training (DAT) was utilized to create speaker representations [15]. Corrective learning was proposed, where the representation of the previous frame is fed to the next input frame [16].

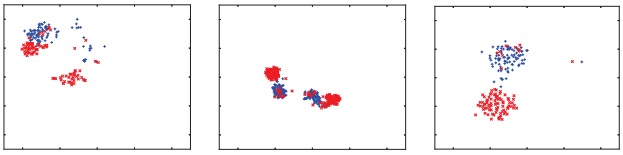
### 2.2. Feature Representation in Redundancy Reduction

**Autoencoder** Autoencoder is a feature learning method that generates bottleneck features. A bottleneck feature is a dimension-reduced form of an input feature, but bottleneck features should contain speaker-specific information. Autoencoder is employed in many fields, including image denoising [17] and generative models [18]. Research on the use of bottleneck feature for speaker representation was carried out on the previous work [19].

**Convolutional Neural Network** Local and global irrelevant factors are discarded by local pooling (or stride) and global average pooling (GAP), thereby capturing identity-aware factors with convolution operations [20].

## 3. DEEP SPEAKER REPRESENTATION

In this paper, we assume that a speaker representation contains unspecified factors. The specified factors of variation are associated with the speaker labels, while the remainders are the unspecified factors of variation [21]. We also assume that the speaker representation can be expressed as a linear combination of latent vectors. Orthogonal Vector Pooling (OVP) is used to estimate these latent vectors, and weights for the latent vectors are estimated with a recombination network.



(a) (b) (c)

**Fig. 2**. t-SNE embedding visualization of two speakers on the VoxCeleb dataset: (a) ResCNN-GAP embeddings, (b) our decomposed vectors, and (c) our recombined embeddings. Red marks correspond to speaker A and blue marks correspond to speaker B.

By using the softmax loss, the weights and the latent vectors are toward learning similar speaker representations from the same speaker dataset. The overall structure of the proposed method is described in Figure 1. The advantage of our method is that it can be easily applied to any DNN because the method is connected at the end of the network. Figure 2 shows a comparison between results on the VoxCeleb dataset [9] from the best previous work (ResCNN-GAP) and our proposed method.

As shown in Figure 2, our method minimizes unnecessary factors of variation so that speaker representations are more discriminative. The detailed approach will be described in Section 3.1 and Section 3.2.

### 3.1. Orthogonal Vector Pooling (OVP)

OVP works as a global pooling method for extracting orthogonal latent vectors. Therefore, it is responsible for performing orthogonal vector decomposition while global variations are reduced. The OVP is shown in Figure 3.

We construct twice the number of final feature maps compared to the conventional CNN architecture in order to decompose a primal representation into two latent vectors, such that the number of feature maps is set to $2d$, where $d$ denotes latent vector dimension. If these latent vectors are learned using orthogonal loss, then they are orthogonal vectors.

$$L_{orthogonal} = \frac{1}{N}\sum_{n=1}^{N}\left|\frac{\boldsymbol{v_1^n}\cdot\boldsymbol{v_2^n}}{\|\boldsymbol{v_1^n}\|_2\,\|\boldsymbol{v_2^n}\|_2}\right| \quad (1)$$

where $v_1^n$ and $v_2^n$ are the decomposed vectors. $n$ is a training

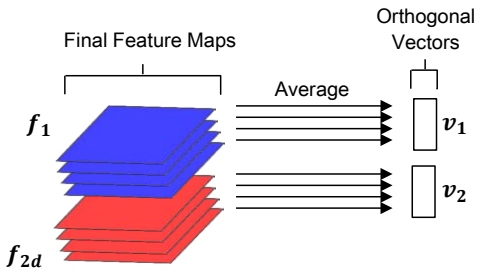



**Fig. 3**. Overview of orthogonal vector pooling

sample index in a mini-batch. The dimension order of the two decomposed vectors is equal to each other because the two decomposed vectors are recombined using an element-wise sum.

The reason for using the orthogonal constraint is to prevent forming similar decomposition vectors which are not worth decomposing. We assume that the decomposed vector weights are restrictively ranged, which facilitates optimal learning in the attention layer. Despite the restricted vector weights, the spanned subspace should be maximized in order to form an overlapped subspace, as shown in Figure 4 (b). The orthogonal loss can help maximize the spanned subspace because the magnitude of the spanned subspace is equal to $|v_1||v_2|\sin(\theta)$ by the cross-product principle. The overlapped subspace enables speaker embeddings to be a global speaker-specific representation, as shown in Figure 4 (c).

### 3.2. Recombination Network

In order to use the decomposed vectors effectively, we propose recombining these vectors using neural networks. The decomposed vectors ($\boldsymbol{v_1}$,$\boldsymbol{v_2}$) were estimated using orthogonal decomposition. Now, we propose a recombination network for recombining these orthogonal vectors and extracting more discriminative speaker representations. We add a fully-connected layer to transform the decomposed vectors ($\boldsymbol{v_1}$,$\boldsymbol{v_2}$) into other decomposed vectors ($f(\boldsymbol{v_1})$,$f(\boldsymbol{v_2})$) for better recombination. The transformed vectors ($f(\boldsymbol{v_1})$,$f(\boldsymbol{v_2})$) are normalized in order to control the vector magnitude with an attention layer as below:

$$\bar{\boldsymbol{v_1}} = \frac{f(\boldsymbol{v_1})}{||f(\boldsymbol{v_1})||_2}, \quad \bar{\boldsymbol{v_2}} = \frac{f(\boldsymbol{v_2})}{||f(\boldsymbol{v_2})||_2} \tag{2}$$

The vector weights to minimize intra-speaker distance and maximize inter-speaker distance using the softmax loss will be estimated using the basic attention layer [7]. Softmax normalization is not included in the attention layer because the attention layer serves as estimating the magnitude of the latent vectors.

$$\alpha_1 = f_a(\boldsymbol{v_1}), \quad \alpha_2 = f_a(\boldsymbol{v_2}) \tag{3}$$

Therefore, we can extract speaker-discriminative representations using a linear combination of the two normalized vectors

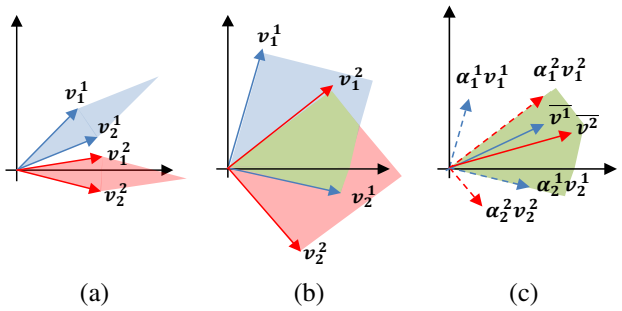



**Fig. 4**. The spanned subspaces and recombined embeddings: (a) decomposed vectors without orthogonal constraint, (b) decomposed vectors with the orthogonal constraint, (c) recombined embeddings with the orthogonal constraint and weights estimated with the attention layer. $\boldsymbol{v_1^1}$ and $\boldsymbol{v_2^1}$ are a pair of decomposed vectors. $\boldsymbol{v_1^2}$ and $\boldsymbol{v_2^2}$ are other pair of decomposed vectors. $\alpha_1^1$, $\alpha_2^1$, $\alpha_1^2$, and $\alpha_2^2$ are vector weights estimated with the attention layer. $\bar{\boldsymbol{v^1}}$ and $\bar{\boldsymbol{v^2}}$ are the recombined speaker embeddings for the same speaker.

($\bar{v_1}$,$\bar{v_2}$) with the estimated weights ($\alpha_1$,$\alpha_2$) in Equation (4), implicitly leading to reduced intra-speaker variability.

$$\bar{\boldsymbol{v}} = \alpha_1 \bar{\boldsymbol{v_1}} + \alpha_2 \bar{\boldsymbol{v_2}} \tag{4}$$

The network ends with a fully-connected layer with softmax.

$$L_{softmax} = -\frac{1}{N}\sum_{n=1}^{N} \log \frac{\exp(\boldsymbol{w_s} \cdot \bar{\boldsymbol{v^n}} + b)}{\sum_{spk} \exp(\boldsymbol{w_{spk}} \cdot \bar{\boldsymbol{v^n}} + b)} \tag{5}$$

in which $w_s$ denotes a correct speaker basis. $n$ is a training sample index in a mini-batch. The entire network can be learned using the following loss.

$$L_{total} = L_{softmax} + \lambda L_{orthogonal} \tag{6}$$

where $\lambda$ is an orthogonality parameter.

## 4. EXPERIMENT

### 4.1. Experiment Setup

The phrase "Hi Bixby" was used as a keyword for a text-dependent dataset. The text-dependent training set consists of anonymized voice search logs and manually collected data. The text-dependent training set comprises about 1.2M utterances from 60k speakers. We use the VoxCeleb dataset [9] for text-independent tasks. The VoxCeleb training set comprises over 100,000 utterances from 1,211 speakers.

For text-dependent tasks, a clean test set was collected from 100 speakers, and a real test set was collected from 20 speakers with real environmental noise. The Voxceleb test set consists of 40 different speakers. The text-dependent test set was cropped to one second to evaluate short utterance. The VoxCeleb test set was used by taking three second crops as

**Table 1**. Network architectures: shortcut connection is added to each pair of 3x3 filters in both ResCNN and OrthResCNN as in [24].

| Layer | ResCNN-GAP | OrthResCNN-OVP (Ours) |
|---|---|---|
| conv1 | $[7 \times 7, 64]$, stride 1 | $[7 \times 7, 64]$, stride 1 |
| conv2 | $[1 \times 1, 64]$, stride 2<br>$[3 \times 3, 64]$ $\times 6$, stride 1 | $[1 \times 1, 64]$, stride 2<br>$[3 \times 3, 64]$ $\times 6$, stride 1 |
| conv3 | $[1 \times 1, 128]$, stride 2<br>$[3 \times 3, 128]$ $\times 6$, stride 1 | $[1 \times 1, 128]$, stride 2<br>$[3 \times 3, 128]$ $\times 6$, stride 1 |
| conv4 | $[1 \times 1, 256]$, stride 2<br>$[3 \times 3, 256]$ $\times 6$, stride 1 | $[1 \times 1, 256]$, stride 2<br>$[3 \times 3, 256]$ $\times 6$, stride 1 |
| pooling | global average pooling<br>$(1 \times 256)$ | orthogonal vector pooling<br>$(1 \times 128, 2)$, orth. loss |
| fc1 | - | $(128 \times 128, 2)$ |
| att1 | - | $(128 \times 1, 2)$ |
| fc2 | (256 / 128$\times$ the number of speakers), softmax loss | |

in [9]. All recordings were converted to single-channel, 16-bit waveforms with 16kHz sampling rate.

We use a Hamming window with 25ms width, 10ms step size, and 512-points FFT. This yields a 64 dimensional log mel-filterbank feature vectors using the librosa library [22]. Feature warping was performed to enforce consistency across the various recording environments [23]. We experiment with $64 \times 100$ for text-dependent tasks and $64 \times 300$ for text-independent tasks. Both tasks were separately trained. We used Adam optimizer with an initial learning rate of $0.001$, and a mini-batch size of 32 for both tasks. Batch normalization and ReLU were applied after each convolution operation. We chose $\lambda = 1.0$ as the orthogonality parameter. The d-vector [6], x-vector [12] and ResCNN-GAP were selected as baselines in our evaluation. ResCNN-FCN is a combination of ResCNN and three FCNs, including a softmax layer without GAP. ResCNN-GAP and OrthResCNN-OVP architectures were constructed by a trivial change of the existing model [24], as specified in Table 1. The speaker model was computed as the average of speaker representations extracted from five enrollment recordings. The cosine distance was used to measure similarity. Performance was assessed by calculating the EER.

### 4.2. Experiment Results

We compared our method with the baselines, and the experimental results are shown in Table 2 and 3. Our method outperforms all baseline systems for both text-dependent and text-independent evaluations. The EER of 1.67% was achieved when our method was used to evaluate text-dependent tasks in real environment. This shows EER reduction of 72.5% and 49.6% relatively compared to the results from the x-vector and the ResCNN-GAP. For text-independent tasks, the proposed approach improves EER from 8.48% to 2.85%, with 66.3% relative error reduction compared to the result from the x-vector method. We report that this result shows better improvement compared to EER of 7.8% [9] and 7.3% [10] obtained with the cosine distance. We also found that CNN-based network yields better results compared to the other networks. Moreover, the use of GAP is helpful to improve performance. It is interesting that CNN architectures provide us to gain performance, by implicitly discarding unspecified factors of variation, with stride and pooling operations.

**Table 2**. EER(%) when evaluating text-dependent short utterances

| Network | Clean Condition | Real Condition |
|---|---|---|
| d-vector [6] | 4.94 | 19.52 |
| x-vector [12] | 1.52 | 6.07 |
| ResCNN-FCN | 1.42 | 5.14 |
| ResCNN-GAP | 1.40 | 3.32 |
| OrthResCNN-OVP | 0.81 | 1.67 |

**Table 3**. EER(%) on VoxCeleb dataset from text-independent test

| Network | VoxCeleb evaluation set |
|---|---|
| x-vector | 8.48 |
| ResCNN-GAP | 5.39 |
| OrthResCNN-OVP | 2.85 |

**Table 4**. EER(%) from ablation experiments for text-dependent test in a real environment.

| Network | Condition | Real Condition |
|---|---|---|
| OrthResCNN-OVP | w/o orthogonal loss | 2.01 |
| | w/o attention layer | 2.05 |
| | w/o transform layer | 2.34 |
| | all modules | 1.67 |

### 4.3. Ablation Experiments

We conducted ablation experiments with a text-dependent dataset. There are three modules for ablation studies, such as orthogonal loss, an attention layer, and a transform layer. The ablation experiments allow us to confirm which module significantly contributes and determine if all modules are necessary. The experiments were performed by investigating one of the three modules. The conditions and results from the ablation experiments are listed in Table 4. The results indicate that each module contributes to an observed performance enhancement. Furthermore, we believe that using orthogonal latent vectors and recombination with estimated weights show a positive effect of leverage on performance.

## 5. CONCLUSION

In this paper, we investigated the deep speaker representation based on orthogonal decomposition and recombination. This method is easy to apply as it can be added to existing CNN architectures, as well as our method can be extended to more than two latent vectors. The proposed approach outperforms the baseline systems and yields a relative EER reduction of 50-70% for text-dependent and text-independent tasks. In future work, we will focus on using recombination network to make the speaker representation more discriminative.

## 6. REFERENCES


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

[2] William M Campbell, Douglas E Sturim, Douglas A Reynolds, and Alex Solomonoff, "Svm based speaker verification using a gmm supervector kernel and nap variability compensation," in *Proc. of ICASSP*. IEEE, 2006, pp. 97–100.

[3] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre 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] Alex Solomonoff, Carl Quillen, and William M Campbell, "Channel compensation for svm speaker recognition.," in *Proc. of Odyssey*, 2004, vol. 4, pp. 219–226.

[5] Andrew O Hatch, Sachin Kajarekar, and Andreas Stolcke, "Within-class covariance normalization for svm-based speaker recognition," in *Proc. of the 9th International Conference on Spoken Language Processing*, 2006, pp. 1471–1474.

[6] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez-Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification.," in *Proc. of ICASSP*. IEEE, 2014, vol. 14, pp. 4052–4056.

[7] FA Chowdhury, Quan Wang, Ignacio Lopez Moreno, and Li Wan, "Attention-based models for text-dependent speaker verification," in *Proc. of ICASSP*. IEEE, 2018, pp. 5359–5363.

[8] Shi-Xiong Zhang, Zhuo Chen, Yong Zhao, Jinyu Li, and Yifan Gong, "End-to-end attention based text-dependent speaker verification," in *Proc. of SLT*. IEEE, 2016, pp. 171–178.

[9] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Proc. Interspeech*, 2017.

[10] Suwon Shon, Hao Tang, and James Glass, "Frame-level speaker embeddings for text-independent speaker recognition and analysis of end-to-end model," in *Proc. SLT*. IEEE, 2018.

[11] Zili Huang, Shuai Wang, and Yanmin Qian, "Joint i-vector with end-to-end system for short duration text-independent speaker verification," in *Proc. of ICASSP*. IEEE, 2018, pp. 4869–4873.

[12] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. of Interspeech*, 2017, pp. 999–1003.

[13] Erik Marchi, Stephen Shum, Kyuyeon Hwang, Sachin Kajarekar, Siddharth Sigtia, Hywel Richards, Rob Haynes, Yoon Kim, and John Bridle, "Generalised discriminative transform via curriculum learning for speaker recognition," in *Proc. of ICASSP*. IEEE, 2018, pp. 5324–5328.

[14] Tian Gao, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "Snr-based progressive learning of deep neural network for speech enhancement.," in *Proc. Interspeech*, 2016, pp. 3713–3717.

[15] Qing Wang, Wei Rao, Sining Sun, Lei Xie, Eng Siong Chng, and Haizhou Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *Proc. of ICASSP*. IEEE, 2018, pp. 4889–4893.

[16] Yandong Wen, Tianyan Zhou, Rita Singh, and Bhiksha Raj, "A corrective learning approach for text-independent speaker verification," in *Proc. of ICASSP*. IEEE, 2018, pp. 4894–4898.

[17] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. of ICML*. ACM, 2008, pp. 1096–1103.

[18] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," in *Proc. of ICLR*, 2014.

[19] Sibel Yaman, Jason Pelecanos, and Ruhi Sarikaya, "Bottleneck features for speaker recognition," in *Proc. of Speaker Odyssey*, 2012, pp. 105–108.

[20] Min Lin, Qiang Chen, and Shuicheng Yan, "Network in network," in *Proc. of ICLR*, 2014.

[21] Michael Mathieu, J. Zhao, P. Sprechmann, A. Ramesh, and Y. LeCun, "Disentangling factors of variation in deep representation using adversarial training," in *Proc. of NIPS*, 2016, pp. 5040–5048.

[22] C. Raffel B. McFee, D. P. Ellis D. Liang, E. Battenberg M. McVicar, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proc. of the 14th python in science conference*, 2015, pp. 18–25.

[23] Jason Pelecanos and Sridha Sridharan, "Feature warping for robust speaker verification," in *Proc. of ISCA Odyssey*, 2001, pp. 213–218.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. of CVPR*. IEEE, 2016, pp. 770–778.