Deep Speaker Representation using Orthogonal Decomposition and Recombination for Speaker Verification

Insoo Kim, Kyuhong Kim, Jiwhan Kim, and Changkyu Choi

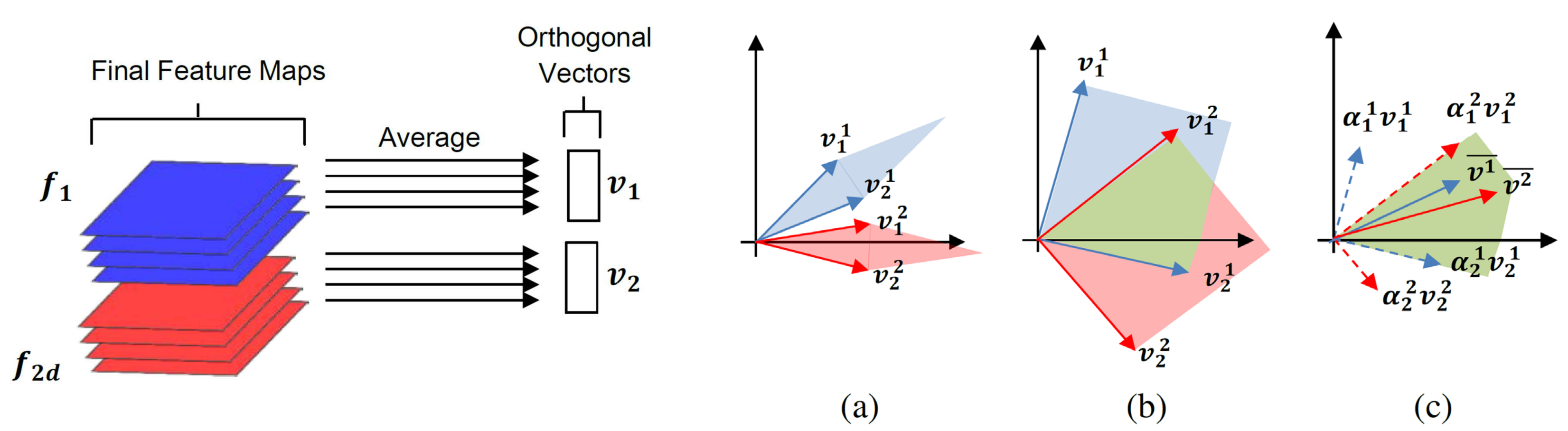
Samsung Advanced Institute of Technology, South Korea

SAMSUNG
SAMSUNG ADVANCED INSTITUTE OF TECHNOLOGY

Main Idea

- The proposed method performs orthogonal decomposition and recombination to obtain the discriminative speaker representation.
- Speaker representation is generated by a linear combination of latent vectors using deep neural network.
- The proposed method can be easily applied to any deep neural network, which is connected at the end of the network.

Orthogonal Decomposition



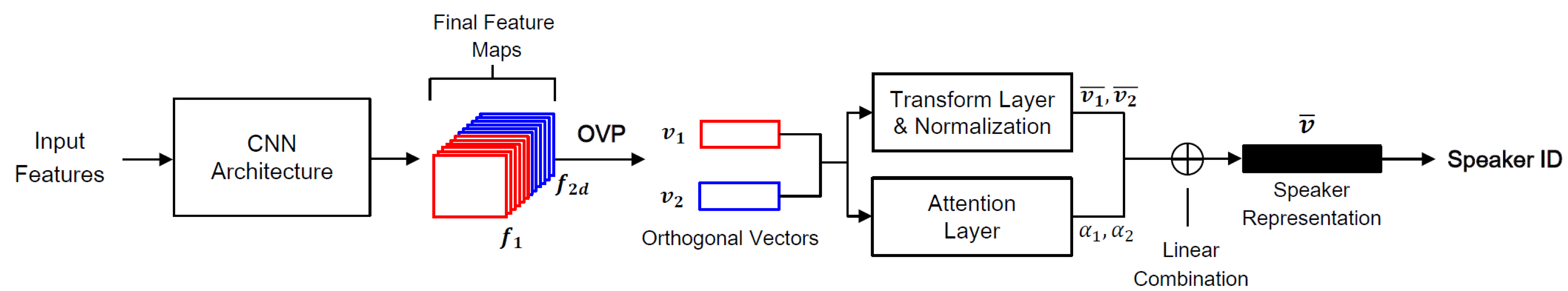
- Construct twice the number of final feature maps compared to the conventional CNN architecture in order to decompose a vector into two latent vectors.
- Orthogonal vectors can be generated by applying the following constraint.

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

Why applying the orthogonal constraint ?

- Prevent forming similar decomposition vectors
- Maximize the spanned subspace so that the overlapped subspace between two examples can be formed.
- The overlapped subspace enables speaker representations to be a global speaker-specific representation.

Network Architecture



Recombination Network

- Transform and normalize the decomposed vectors to control the vector magnitude

$$\boldsymbol{\bar{v}_1} = \frac{f(\boldsymbol{v_1})}{||f(\boldsymbol{v_1})||_2}, \quad \boldsymbol{\bar{v}_2} = \frac{f(\boldsymbol{v_2})}{||f(\boldsymbol{v_2})||_2}$$

- Estimate magnitude of decomposed vectors using attention layer toward minimizing intra-speaker distance and maximizing inter-speaker distance using the softmax loss

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

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

- Be able to extract speaker-discriminative representation using a linear combination of the two normalized vectors with the estimated amplitudes

$$\boldsymbol{\bar{v}} = \alpha_1\boldsymbol{\bar{v}_1} + \alpha_2\boldsymbol{\bar{v}_2}$$

- The entire network can be learned using the following loss.

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

Experimental Result

- Input : logfbank features (input feature map : 64 x 100, 64 x 300)
- Text-dependent task : In-house dataset (“Hi, bixby”)
- Text-independent task : VoxCeleb dataset

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 $[3\times 3, 64]\ \times 6$, stride 1	$[1\times 1, 64]$, stride 2 $[3\times 3, 64]\ \times 6$, stride 1
conv3	$[1\times 1, 128]$, stride 2 $[3\times 3, 128]\ \times 6$, stride 1	$[1\times 1, 128]$, stride 2 $[3\times 3, 128]\ \times 6$, stride 1
conv4	$[1\times 1, 256]$, stride 2 $[3\times 3, 256]\ \times 6$, stride 1	$[1\times 1, 256]$, stride 2 $[3\times 3, 256]\ \times 6$, stride 1
pooling	global average pooling (1×256)	orthogonal vector pooling $(1\times 128, 2)$, orth. loss
fc1	-	$(128\times 128, 2)$
att1	-	$(128\times 1, 2)$
fc2	(256 / 128× the number of speakers), softmax loss	

Table 1. Network Architectures

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 2. EER (%) on text-dependent task

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

Table 3. EER (%) on text-independent task

Conclusions

- The proposed approach outperforms the baseline systems and yields a relative EER reduction of 50-70% for text-dependent and independent tasks.
- The better performance can be achieved when we apply our method to CNN architectures.
- This method can be extended to more than two latent vectors.
- Our method can be applied to other applications, such as face verification, image classification, and sound classification.