Co-compressing and Unifying Deep CNN Models for Efficient Human Face and Speaker Recognition

Timmy S.T. Wan[1,2], Jia-Hong Lee[1,2], Yi-Ming Chan[1,2], and Chu-Song Chen[1,2]

[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan,
Email: {timmywan, honghenry.lee, yiming, song}@iis.sinica.edu.tw
[2]MOST Joint Research Center for AI Technology and All Vista Healthcare

Abstract

Deep CNN models have become state-of-the-art techniques in many application, e.g., face recognition, speaker recognition, and image classification. Although many studies address on speedup or compression of individual models, very few studies focus on co-compressing and unifying models from different modalities. In this work, to joint and compress face and speaker recognition models, a shared-codebook approach is adopted to reduce the redundancy of the combined model. Despite the modality of the inputs of these two CNN models are quite different, the shared codebook can support two CNN models of sound and image for speaker and face recognition. Experiments show the promising results of unified and co-compressing heterogeneous models for efficient inference.

1. Introduction

Face recognition (FR) and speaker recognition (SR) are both important modules for applications such as access control, human/robotic interaction, and multimedia systems. Deep learning techniques have been shown promising to improve the FR and SR performance in recent years. However, most deep learning models are developed for either FR or SR tasks. In this paper, we present a deep convolutional neural network (CNN) approach that can jointly perform FR and SR in a single neural-network model. Besides, with our approach, both FR and SR tasks can be realized in one compressed neural network, and thus the storage and execution time required for the multimodal inference can be reduced.

To train a multi-task model in deep learning, a typical approach is to construct a CNN architecture with multi-task outputs in the final classification layer at first and then train this model with the union of training data from all tasks. However, such an approach requires a tedious trial-and-error procedure because the architecture chosen could be inappropriate in the beginning and multiple re-training processes are needed in every trial. Even if neural network architecture search (NAS) [29] techniques can find appropriate architectures, it still requires taking a long time and consuming a lot of computational resources to generate satisfiable multi-task models.

One possible approach is to leverage on existing deep CNN models already trained for an individual FR or SR task. For example, with fast-growing deep CNNs, FR has gotten great performance improvement nowadays. Schroff et al. [20] propose triplet loss and design a network structure for FR. Liu et al. [12] introduce additive angular margin loss and redesign the network structure for FR. These models are publicly available with high accuracy on large face datasets (such as the LFW dataset [10]). Leveraging on well-performed single-task models has the advantage of preserving the recognition accuracy of one modality more easily. Nevertheless, it is still non-trivial to merge two well-performed models without compromising the performance of the individual task.

In this paper, we present an approach that leverages on well-trained individual models of both FR and SR. We then merge the two single-modality models into a unified one while keeping the compactness of the merged model for multi-modal inference. When deep CNN models are learned, they often have much redundancy in the network weights, and thus the models can be compressed before deployment for inference [5, 27, 7]. In our work, we not only merge the two models but also "co-compress" them into a single model, and thus the resulted model is compact (with a smaller model size) and more suitable for efficient multi-modal inference.

Our approach follows the principle of NeuralMerger [2]. In this technique, two merged layers share a common codebook consisting of a set of codewords; the individual-task

weights are obtained by looking up the codebook, where its codewords are differentiable and can be fine-tuned through back-propagation algorithm in an end-to-end manner. In our study, although the modalities are different (one is face image and the other is human voice), we find that there are still join-redundancy among their model weights. The codeword-sharing ratio is getting increased when the convolutional layer is deeper. It reveals that, despite the two signal sources (image and sound) differ larger in early layers, after processing them with several previous layers, the obtained middle representations become more common and thus can be processed with a higher ratio of shared weights in later deeper convolutional layers. The final merged model still maintains comparable performance to that of the original models on both FR and SR tasks. The model is compact (with only 40% of the individual-task model size), and thus more suitable for inference on resource-limited devices.

The rest of this paper is organized as follows. In Section 2, we briefly review recent FR, SR, and multi-modal learning approaches. In Section 3, we introduce our approach that includes the CNN model construction and learning for FR and SR, as well as co-compressing the two models and unifying them into a single model. Experimental results are presented in Section 4. Finally, a conclusion is given in Section 5.

2. Related Work

In this section, we briefly review face recognition, speaker verification, and multimodal learning.

2.1. Face Verification

To discriminate whether two faces are the same, recent approaches concentrate on two directions. The first is to improve the loss function to extract better facial features [13] [25]. Liu et al. [13] propose a generalized large-margin softmax (L-Softmax) loss function. It can enhance intra-class compactness and inter-class separability between facial features by adjusting the desired margin.

The other direction is to redesign the structure of convolutional neural networks with innovative loss functions to enhance the discriminability, such as Centerloss [26], FaceNet [20], and SphereFace [12]. In Centerloss [26], the network structure is composed of three convolutional layers, three local convolutional layers, PReLU [6] activations, and one fully-connected layer. They utilize center-loss and softmax loss to enhance the discriminability. The center loss function focuses on the center for facial feature vector distribution of each class and minimizes the distances between facial feature vectors and their corresponding class center. In FaceNet [20], their network structure is composed of eleven Inception blocks and one fully-connected layer. They use triplet loss to optimize the face embedding by keeping positive pairs closer and negative pairs far from each other. However, the triplet picking procedure is time-consuming. In SphereFace [12], the network structure is composed of twenty convolutional layers with shortcuts and PReLU activations. They leverage the angular softmax (A-Softmax) loss function to amplify the margin between the target identity and the non-target identity.

We refer to SphereFace [12] as our baseline network architecture due to its good performance and ease of implementation.

2.2. Speaker Verification

Before emerging of deep learning applications, i-vector technique [3] was widely used in speaker verification task. After that, researchers devoted to learning a speaker embedding with deep neural networks in an end-to-end manner. A deep neural network approach, D-vector [21], learns a frame-level embedding from the average of outputs of last hidden layer as a target speaker model. Various new network architectures have been proposed recently. Most works [15] [9] convert each frame to a spectrogram as an image feeding to a 2-D convolutional neural network. In contrast, [16] constructs a 1-D convolutional neural network to train directly with the raw audio signals. Besides, similar to the progress of face recognition, two works [9, 11] introduce revised loss criteria to learn deep speaker features. To keep the speaker embedding robust to noise, the work in [9] combines triplet loss with an intra-class loss to minimize intra-class variations. To achieve a stronger discriminability, Li et al. [11] learn angularly discriminative features using A-Softmax [12]; the results are as good as A-Softmax loss function in the face verification task.

We also build a 2-D convolutional network for SR, which is the same as the one we use in the FR task. Similarly, we optimize the network with A-Softmax loss criteria as well.

2.3. Multimodal learning

Multimodal learning can be summarized into two branches of study. The first one is the fusion-based approach. In [14], the authors construct a multimodal CNN through concatenating two heterogeneous features at feature-level. In Vegrad [22], they train two distinct models and fuse the output of those at decision-level. To determine the best fusion scheme, Vielzeuf et al. [23] propose an easy modification to most existing neural network models and the parameters used in fusion become learnable. Although the mentioned approaches are easy to implement, the complexity of the neural network model is increased and requires double or more memory space for inference.

The other direction is to build a cross-domain multi-tasking model. An intuition way is to construct a single network to handle different domains. To achieve this goal, a unified model is designed in [8] to handle multiple tasks across domain using universal data representation. How-

Layer	Conv1.x	Conv2.x	Conv3.x	Conv4.x	FC1
Block Structure	$3\times3, 64, S2$ $\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times1$	$3\times3, 128, S2$ $\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times2$	$3\times3, 256, S2$ $\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times4$	$3\times3, 512, S2$ $\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times1$	512

Figure 1. The network architecture of CNN-20. The workflow is from left to right, top to bottom. Conv1.x, Conv2.x, Con3.x, and Conv4.x contain the convolution units. Square brackets represent residual blocks. FC1 is the fully connected layer.

ever, it may require strenuous trial-and-error in the model selection process, given a huge amount of combined training data.

To tackle the model size and time-consuming issues, our approach merges well-trained FR and SR models through joint compression. We not only remove the co-redundancy between models but also avoid the difficulty of the model selection and trial-and-error process.

3. Method

In this paper, we use the CNN-20 architecture [12] to construct both our FR and SR deep-CNN models. Figure 1 gives an architecture description of it, which consists of four blocks (20 layers) of convolutional layers and a fully connected layer. The squared brackets express the residual block structure with shortcut connections.

3.1. FR Model Construction

In this work, we train the CNN-20 network with A-softmax loss on VGGFace2 dataset to construct our FR model. VGGFace2 dataset is larger than CASIA-WebFace dataset employed in [12]. Each input RGB image is pre-processed by using a face detector, MTCNN [28], to crop the face region inside the image. The cropped region is then resized to 112×112 for both training and testing.

On standard face verification benchmark, LFW, our implemented FR model of the CNN-20 architecture can achieve 99.42% accuracy (higher is better), which is the same as that reported in [12] with a deeper architecture (64 layers).

3.2. SR Model Construction

As A-softmax loss is capable of learning more angularly discriminating feature embedding and has demonstrated its effectiveness in FR, this idea is recently employed in SR [11] for learning speaker embedding from deep CNNs as well. In our work, we use the CNN-20 architecture with A-softmax loss to train the SR model too. The dataset employed for SR training is Voxforge dataset[1], an open source speech corpus collected transcribed speech data from volunteer speakers. Each sample is converted to log-power mel-scaled spectrogram, with the length of the FFT window being 2048, the number of samples between successive frames being 69. The spectrogram is then cropped to 112×112.

On the evaluation data of Voxforge dataset, our SR model can achieve the half total error rate (HTER) of 1.86% (lower is better). Our model outperforms the i-vector (cosine distance) and i-vector (PLDA) approaches, which achieve 2.82% and 5.87% HTERs, respectively.

Although [16] is slightly better than our baseline model (1.2% HTER), the performance is evaluated on the setting that every speaker has his (or her) own CNN model. This one-CNN-per-person setting can increase the performance, but needs multiple CNN models for multiple speakers; whereas a new model is required to be built for a newly registered speaker. Our approach uses a single CNN model that extracts the feature embedding per speaker. Speaker verification and identification can be easily performed by comparing the distance between the embedding of speakers and nearest-neighbor-search in the embedding space, respectively. Speaker verification or recognition can thus be performed more efficiently than that in [16] when the number of users is increased.

3.3. Merging FR and SR Models

The FR and SR models are co-compressed and merged to form a unified F&SR model in our work. This is unlike previous approaches [19] often combining two models with newly added structures, such as bridging layers or common embedding between the hidden layers of two networks, so that two modalities can be simultaneously executed in a single network. Despite the performance could be improved by combining the two modalities with newly added structures, the resulted network architecture is often more complex and thus cannot be realized in edge devices easily. Our approach employs the joint redundancy between the two well-learned networks, so that they are co-compressed to form a more compact model, which is more suitable to serve for inference on resource-limited devices.

Some approach (e.g., Liu et al. [14]) assumes that the signal sources of the two modalities are received synchronously. That is, the human face images and their voices should be gotten simultaneously and are co-used for human identification. The two modalities are helpful to each other and can be used to jointly validate the results so that the recognition accuracy can be boosted. However, sometimes only one type of signal is received in real applications. For example, in a home-robot system, we would demand to identify a person's voice when he (or she) stands behind the robot, i.e., out of sight of the camera; we could also need to identify the human face when he (or she) doesn't utter a sound. Our approach supports both the synchronous and asynchronous modes, which does not assume the availability of both modalities for person identification. We focus on co-compressing the two well-trained models so that

[1] http://www.voxforge.org/

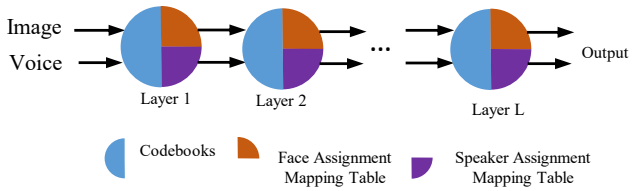

Figure 2. The merged model of the face and speaker recognition. Both image and voice share the same codebooks to reduce the redundancy. Two assignment mapping tables manage each mode independently.

their performance can be preserved and flexibly applicable for both asynchronous and synchronous modes on resource-limited devices. In the asynchronous mode, the input from each mode is processed independently. On the other hand, in the synchronous mode, our approach obtains the recognition results from both modalities. They could be further refined with a post-fusion (e.g. soft voting on either embedding or classification layers) to obtain a better final result for the synchronous mode, but this is out of the main focus of this work.

Given the FR and SR models of the same architecture (CNN-20) that contains 20 convolutional layers and 1 fully connected layer as shown in Figure 2, we merge the corresponding layers between them into a single layer. In the following, we present the approach of merging a pair of convolutional layers, one from the FR model and the other from the SR model. The same principle is then followed when merging fully connected layers.

Let $\mathbf{L}^F$ and $\mathbf{L}^S$ be a pair of the convolutional layers to be merged, where $\mathbf{L}^F$ is from the FR model and $\mathbf{L}^S$ is from the SR model, respectively. Assume that the input and output tensors of the respective layer is of dimension $M \times N \times D$ and $M \times N \times P$, respectively, where M, N are the spatial size, D is the depth (or number of channels) of the input tensor, and P is the depth of the output tensor. The convolution kernel applied to this layer is of size $m \times n \times D \times P$, where m, n are the spatial size, D is the depth, and P is the number of the kernel. Without loss of generality, assume that m, n are odd numbers with $m = 2w + 1$ and $n = 2h + 1$.

In the principle of NeuralMerger [2], convolution kernels of the two layers to be merged are jointly represented by fewer codewords, so that the two layers can be merged to form a more compactly represented layer. The merged model can take advantage of these fewer codewords to construct lookup tables for efficient inference with negligible accuracy drop. We briefly review NeuralMerger as follows. The convolution kernel is divided into $1 \times 1 \times r$-dimensional subspaces, where r is along the depth direction. There are K such subspaces for one model, where $K = \lceil D/r \rceil$. In each subspace, there are $T = mnP$ kernel segments of dimension r for one model. Because we have two models, FR and SR, there are $2T$ kernel segments of dimension r in each subspace.

For each dim-r subspace, NeuralMerger finds C codewords to jointly encode the $2T$ kernel segments of the two models, $C < T$. The codewords are initially found via vector quantization using K-means algorithm. Denote the C clustering centers (i.e., codewords) found for the k-th subspace to be $\{b_{c,k} \in \mathcal{R}^r | c = 1 \cdots C, k = 1 \cdots K\}$. Each of the $2T$ kernel segments is then assigned to one of the C clustering centers. These fewer codewords are used to form lookup tables for efficient inference.

Here, how to use these codewords to reconstruct the original convolution operation is explained. We define the codeword assignment mapping of clustering to be $\pi^F(i_0, j_0, p, k)$ and $\pi^S(i_0, j_0, p, k)$ for the FR and SR models, respectively, where $-w \le i_0 \le w$ and $-h \le j_0 \le h$ represent the (i_0, j_0)-th spatial elements of kernel, $p \in \{1 \cdots P\}$ represents the p-th kernel, and $k \in \{1 \cdots K\}$ represents the k-th subspace. The convolution operation in the layer of the FR model can then be approximated as

$$y^F_{i,j;p} = \sum_{k=1}^{K} \sum_{i_0=-w}^{w} \sum_{j_0=-h}^{h} \langle x^F_{i+i_0,j+j_0;k}, b_{\pi^F(i_0,j_0,p,k);k} \rangle, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner-product of r-dimensional vectors, $y^F_{i,j;p}$ is the output at the spatial location (i, j) of the p-th channel, $x^F_{i,j;k}$ is the input of spatial location (i, j) of the k-th subspace, $1 \le i \le M$ and $1 \le j \le N$. Please note that the inner-product is conducted with fewer codewords to form lookup tables. Then the convolution is done by sum up values in the lookup tables using (1). Likewise, for the SR model, the convolution-layer operation is approximated as

$$y^S_{i,j;p} = \sum_{k=1}^{K} \sum_{i_0=-w}^{w} \sum_{j_0=-h}^{h} \langle x^S_{i+i_0,j+j_0;k}, b_{\pi^S(i_0,j_0,p,k);k} \rangle. \tag{2}$$

In the convolutional layer, inner products (or $1 \times 1 \times r$ convolutions) of the input and the codewords are performed in a subspace. Because C, the number of codewords jointly representing the kernel coefficients of the FR and SR models, is often significantly smaller than that of the kernel segments T in that subspace of one model, the number of inner-product operations is reduced in the mutually encoded convolutional layer. Hence, the layers can be jointly compressed with a smaller size and faster inference speed.

The codewords co-used in NeuralMerger removes the joint redundancy of the two well-trained models. Although the codewords $\{b_{c,k} | c = 1 \cdots C, k = 1 \cdots K\}$ are initialized via vector quantization of the original models, Chou et al. [2] show that the codewords are differentiable and thus can be fine-tuned from data through back-propagation procedure (referred to as calibration training in [2]). In the

calibration training of Neural Merger, the codewords are re-trained with two combined error terms. One is the loss functions utilized in the original FR and SR models, which fulfill the goals of high FR and SR accuracy. The other is the layer-wise output mismatch error, which follows the principle of student-teacher network of distilling training, where the co-compressed model approximates the outputs of the original well-trained FR and SR models at every layer.

In the above, co-compression of convolutional layers by joint encoding is depicted. Merging fully-connected layers follows the same principle in NeuralMerger, where the weights are divided into dim-r subspaces. The codewords are found to jointly represent the weights in each subspace and fine-tuned with calibration training in the fully-connected layers as well.

To realize our approach, we use GPUs (Nvidia) to train the individual FR and SR models, and also for the calibration training of the merged model. We use CPU in the inference stage so that the results are more generalizable to edge devices that may not contain GPUs.

4. Experiments

In this section, we firstly introduce datasets with evaluation metrics for each respective task and then describe the preprocessing procedure with implementation details as well as our baseline. Next, we present the different settings of our merged model and make a comparison between ours and baseline in terms of accuracy, speed and compression ratio.

4.1. Dataset and Metrics

In the face verification task, we train on VGGFace2 [1] and evaluate the performance on LFW dataset. The details of both are as follows:

VGGFace2 dataset [1] is a large facial image dataset consisting of 3.31 millions images of 9,131 identities collected from online search engine and the images exhibit large variations in pose, race, age, occlusion, gender. In this setup, we use the official training set of 8,631 identities as our training data.

LFW dataset [10] contains more than 13,233 images of faces and 5,749 subjects collected from the web. In order to test the effectiveness of the learned model, the images are arranged as a 6,000-pair verification task and we can follow the ten-folds cross-validation protocol to evaluate it.

In the speaker verification task, we perform an experiment on 300 speakers chosen from VoxForge. The description of the dataset is as follows:

VoxForge[2] is an open source speech corpus collected transcribed speech data from volunteer speakers. We follow the same settings from [16] to select 300 speakers from the database. Each speaker contains at least 20 utterances recorded at 16 bit, 16kHz in a clean environment. From chosen speakers, the dataset is further divided into three subsets: the training set, the development set, and the evaluation set and each of them contains 100 speakers respectively. To evaluate the performance of the learned model, a single threshold value is predetermined to meet an Equal Error Rate (EER) in the development set. Then, we calculate the Half Total Error Rate (HTER) using the chosen threshold. As compared to the results in [16], we also present the performance in terms of the HTER.

[2] http://www.voxforge.org/

4.2. Data Preprocessing

In the face verification task, we first detect the face using MTCNN [28], which can generate location of the face and five facial landmarks, including right eye, left eye, nose, right mouth corner and left mouth corner. Then, We apply the affine transform to align the face based on the standard location of five facial landmarks and resize the facial images to become 112×112. For data augmentation, we apply normalization with a mean of 0.5 and a standard deviation of 0.5 in the testing phase. In the training phase, we further apply random horizontal flip as well as normalization mentioned before.

In the speaker verification task, The incoming speech data is split into several small chunks of 510ms with 50% overlap. For each small chunk, we compute the log-mel spectrogram using librosa audio processing library, with the FFT window length of 2048, hop length of 69, and 112 mel-bands. Then, we crop the generated log-mel spectrogram into 112×112 and then copy it thrice as a kind of three-channel image. In addition, we also apply normalization with a mean of 0.5 and a standard deviation of 0.5 for data augmentation in both training and testing phase.

4.3. Implementation Details

Baseline. For both tasks, we obtain two pretrained networks as our baseline using A-Softmax loss. Each network architecture we adopt is CNN-20 trained over 40 epochs with the batch size of 256 and the network parameters are optimized using stochastic gradient descent with an initial learning rate of 0.01, a momentum of 0.9 and a weight decay of $5 \cdot e^{-4}$. Furthermore, the learning rate scheduling is also applied to decrease the learning rate by 0.1 at epoch 20, 30, 36 respectively. In the face verification task, we train on the images of top-4001 class over the 40 epochs. Then, we finetune on the remaining images over the 40 epochs. In the speaker verification task, we train on the whole training set over 40 epochs. Before merged, CNN-20 can achieve 99.42 accuracy on LFW benchmark and 1.86% HTER on the evaluation set on VoxForge.

To address the approach about embedding calculation,

the embedding can be easily obtained from the fully connected layer activation using different input modalities. However, to construct a single speaker embedding, post-processing is required due to numbers of input data from the target speaker. To handle this, we extract the mean embedding from log-mel spectrograms of the target speaker. As a result, we are able to measure the similarity between a pair of embeddings using cosine similarity in both verification tasks.

Merging Face and Speaker CNN. We define two parameters to control the size of the merged model. The first one is the dimension of the subspace, r, and the other one is the number of codewords in a subspace, C. The following settings are shown in Table 1. By default, we set the number of codewords to be $C = 256$ for all layers.

After seeking a set of representative codewords to merge, we recover the accuracy of the separately well-trained models into a merged quantized one using end-to-end calibration training. We finetune the codewords using a combination of A-Softmax (L_{ang}) and distilling loss ($L_{distilling}$), as shown in (3), where λ is the combination coefficient.

$$L_{total} = L_{ang} + \lambda \cdot L_{distilling}. \tag{3}$$

The distilling loss defined in (4) computes the one-norm differences of the block-wised outputs, where x_l is the input, $y(x_l)$ is the output of merged model, $f(x_l)$ is the output of the well-trained model and l indexes the blocks in CNN-20.

$$L_{distilling} = \sum_{l} |f(x_l) - y(x_l)|,\ l \in [1, 5]. \tag{4}$$

The coefficient for the loss term in Eq. (3) is set to $\lambda = 10$. We train the model in an alternative manner over 60 epochs with the batch size of 256 via PyTorch [17].

4.4. Results

After finding the codewords for joint representation of the two models via vector quantization, we use 20% training data for the calibration training. The model merging results are summarized in Tables 2 and 3. In the tables, five configurations of different settings of subspace dimensions (**MergerA-E** in Table 1) achieve model-size compression ratios from 1.5x to 5.2x as shown in Table 3. When the subspace dimensions become larger, the merged model size is getting smaller, thus resulting in a higher compression ratio.

Compared to the performance of the original FR model (99.42% accuracy) and SR model (1.86% HTER), the performance of the merged models drops only a little (from 0.17% to 0.59% for FR and 0.58% to 0.68% for SR), and quite satisfiable results can still be obtained by our approach. Compared to the previous individual approaches on FR and SR, our merged models achieve an overall accuracy from 98.83% to 99.25% for FR and HTER 2.44% to 2.54% for SR, respectively. The performance is roughly comparable to that of the previous FR and SR approaches when a compact and unified model is utilized in our approach. Although 1-D CNN [16] can obtain higher performance, multiple speaker-dependent models are used, which is impractical for real applications as discussed in Section 1.

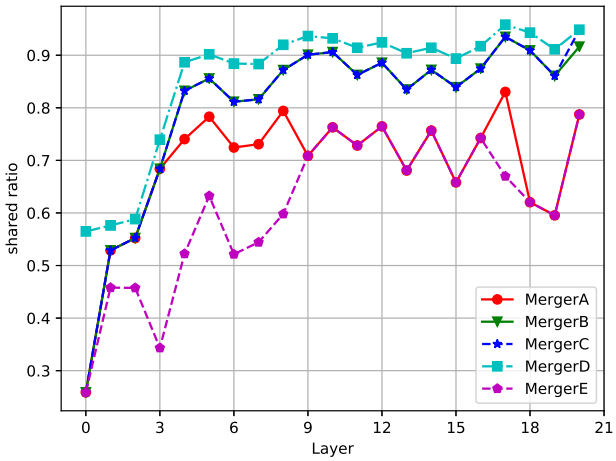

Figure 3. Codeword-sharing ratio of each layer in our all merged models.

Figure 3 shows the layer-wise codeword-sharing ratio in the merged model, which computes how frequent the codewords are shared by both FR and SR in the inference stage of the merged model. It can be seen that, for all configurations **A**-**E**, the codeword-sharing ratio is very small in the starting layer. This would be because that image and voice are heterogeneous signal sources, the convolution weights able to be shared are few in the beginning. Nevertheless, after forwarding several layers, the ratio of sharing is increased. In the end, the ratio remains roughly stable in deeper layers. This phenomenon reveals that the merged model can find feature representations commonly useful for both FR and SR in several layers, despite the modalities are quite different. The deeper layers can then jointly handle the remaining inference based on the common representations found. Hence, the obtained middle representations in our merged model become more common and can be processed with a higher ratio of shared weights in later deeper layers. The final merged model still maintains comparable performance to that of the original FR and SR models.

Besides reporting the accuracy as compared to the existing approach, we also provide an overall performance analysis in compression ratio and speedup. As shown in Table 3, the single merged model size can be compressed up to five times with only a negligible accuracy drop among different parameter settings. When comparing with MergerB and MergerC, we observe the subspace dimension (r) in the fully connected layer has a large influence on the compression ratio. In addition, we find that higher compression

Table 1. The settings of r in each layer.

Para	Conv1.x			Conv2.x					Conv3.x									Conv4.x			FC1
MergerA	3	8	8	8	16	16	16	16	16	32	32	32	32	32	32	32	32	32	64	64	64
MergerB	3	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
MergerC	3	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	4
MergerD	1	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
MergerE	3	16	16	32	32	32	32	32	32	32	32	32	32	32	32	32	32	64	64	64	64

Table 2. The accuracy of each method in our experiment.

Methods	Face	Speaker
	LFW Acc.(%)	HTER
UBM-GMM [18]		3.05
i-vector+cosine distance [4]		2.82
ISV [24]		2.4
CNN-20 (Ours)		1.86
1-D CNN [16]		1.2
FaceNet [20]	99.55	
CNN-20 (Ours)	99.42	
MergerA	**98.83**	**2.54**
MergerB	**99.18**	**2.44**
MergerC	**99.1**	**2.49**
MergerD	**99.25**	**2.51**
MergerE	**98.85**	**2.5**

Table 3. The overall performance of our merged model.

Param.	Compr.	Accuracy Drop (%)	
		Face	Speaker
MergerA	5x	0.59	0.68
MergerB	2.48x	0.24	0.58
MergerC	1.87x	0.32	0.63
MergerD	1.5x	0.17	0.65
MergerE	5.2x	0.57	0.64

ratio does not directly reflect the higher accuracy drop i.e. 0.24% drop in MergerB vs. 0.32% in MergerC. Therefore, using a proper r to reduce co-redundancy perhaps bring a good improvement in accuracy and model size. On the other hand, we further examine the inference speed of MergerE. Instead of estimating our model speedup with theoretical FLOPS ratio, we verify the speedup ratio using CPU with C++ implementation and BLAS library. In the Intel CPU i5-4570 system in single thread mode, the speedup ratio of the merged system is up to 1.50. In a single thread mode of ARM A57 CPU, the speedup ratio of the integrated system is about 2.08 times faster. We argue that it is due to the memory access cost is higher in ARM than the x86 system since we have compressed the model 5.2 times smaller.

5. Conclusion

In this paper, we present a unified deep-learning model for multimodal FR and SR. Well-trained individual models are incorporated to produce the co-compressed merged model, and the performance can be restored with part of the training set. The experiments show that even the inputs are from different modalities, the merged model maintains the performance of FR and SR well, and is more compact and suitable for the inference on resource-limited devices. We also show that the merged model can enforce the common feature representations of FR and SR in the early layers and gradually increases the codeword-sharing ratio in the merged model. The future work will be integrating and co-compressing convolutional networks and recurrent networks for video-based FR and SR.

References

[1] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 67–74, 2018.

[2] Y.-M. Chou, Y.-M. Chan, J.-H. Lee, C.-Y. Chiu, and C.-S. Chen. Unifying and merging well-trained deep neural networks for inference stage. In *IJCAI*, 2018.

[3] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19:788–798, 2011.

[4] 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*, 19(4):788–798, 2011.

[5] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *ICLR*, 2016.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[7] Y. He, G. Kang, X. Dong, Y. Fu, and Y. Yang. Soft filter pruning for accelerating deep convolutional neural networks. In *IJCAI*, 2018.

[8] L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit. One model to learn them all. *CoRR*, abs/1706.05137, 2017.

[9] N. Le and J.-M. Odobez. Robust and discriminative speaker embedding via intra-class distance variance regularization. In *Proceedings of Interspeech*, 2018.

[10] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua. Labeled faces in the wild: A survey. In *Advances*

in face detection and facial image analysis, pages 189–248. Springer, 2016.

[11] Y. Li, F. Gao, Z. Ou, and J. Sun. Angular softmax loss for end-to-end speaker verification. In *International Symposium on Chinese Spoken Language Processing*, 2018.

[12] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6738–6746, 2017.

[13] W. Liu, Y. Wen, Z. Yu, and M. M. Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016.

[14] Y.-H. Liu, X. Liu, W. Fan, B. Zhong, and J.-X. Du. Efficient audio-visual speaker recognition via deep heterogeneous feature fusion. In *Chinese Conference on Biometric Recognition*, pages 575–583. Springer, 2017.

[15] E. Malykh, S. Novoselov, and O. Kudashev. On residual cnn in text-dependent speaker verification task. In *SPECOM*, 2017.

[16] H. Muckenhirn, M. Magimai-Doss, and S. Marcel. Towards directly modeling raw speech signal for speaker verification using cnns. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4884–4888, 2018.

[17] 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 *NIPS-W*, 2017.

[18] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

[19] S. Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[20] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

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

[22] S. Vegad, H. Patel, H. Zhuang, and M. Naik. Audio-visual person recognition using deep convolutional neural networks. *J. Biom. Biostat.*, 2017.

[23] V. Vielzeuf, A. Lechervy, S. Pateux, and F. Jurie. Centralnet: A multilayer approach for multimodal fusion. In *ECCV Workshops*, 2018.

[24] R. Vogt and S. Sridharan. Explicit modelling of session variability for speaker verification. *Computer Speech & Language*, 22(1):17–38, 2008.

[25] F. Wang, J. Cheng, W. Liu, and H. Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25:926–930, 2018.

[26] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

[27] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng. Quantized convolutional neural networks for mobile devices. *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4820–4828, 2016.

[28] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23:1499–1503, 2016.

[29] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. *ICLR*, 2017.