Local Classifier Chains for Deep Face Recognition

Lingfeng Zhang, Ioannis A. Kakadiaris
Computational Biomedicine Lab
University of Houston
{lzhang34, ioannisk}@uh.edu

Abstract

This paper focuses on improving the performance of current convolutional neural networks in face recognition without changing the network architecture. We propose a hierarchical framework that builds chains of local binary neural networks after one global neural network over all the class labels, Local Classifier Chains based Convolutional Neural Networks (LCC-CNN). Two different criteria based on a similarity matrix and confusion matrix are introduced to select binary label pairs to create local deep networks. To avoid error propagation, each testing sample travels through one global model and a local classifier chain to obtain its final prediction. The proposed framework has been evaluated with UHDB31 and CASIA-WebFace datasets. The experimental results indicate that our framework achieves better performance when compared with using only baseline methods as the global deep network. The accuracy is improved by 2.7% and 0.7% on the two datasets.

1. Introduction

Visual recognition is one of the hottest topics in the fields of computer vision and machine learning. In recent years, many deep learning models have been built to set the new state-of-the-art results in image classification, face recognition, and many other visual recognition tasks [18, 7, 26]. Among these tasks, most of the breakthroughs were achieved with deep Convolutional Neural Networks (CNN) [12]. Krizhevsky *et al.* [11] proposed the classic eight layer CNN model (AlexNet) with five convolutional and three fully connected layers. The model is trained via back-propagation through layers and performs extremely well in domains with a large amount of training data. Since then, many new CNN models have been constructed with larger sizes and different architectures to improve performance.

Increasing deep neural network size includes increasing the number of levels and the number of units of each level [21]. Simonyan *et al.* [21] explored the influence of CNN depth by an architecture with small convolutional filters (3×3). They achieved a significant improvement by pushing the depth to 16-19 layers. Szegedy *et al.* [25] introduced GoogLeNet as a 22-layer Inception network, which achieved impressive results in both classification and detection tasks. He *et al.* [9] proposed Residual Networks (ResNet) with a depth of up to 152 layers, which set new records for many visual recognition tasks. Furthermore, the authors released a residual network of 1K layers with new improvements. A larger model means a larger number of parameters. To reduce the storage requirement of the parameters, Han *et al.* [8] introduced "deep compression" by combining pruning, quantization, and Huffman encoding to avoid sacrificing performance. A larger model also dramatically increases computational complexity. The training process is accelerated with multiple GPUs and distributed deep network techniques [2]. Methods for accelerating test-time computation of CNNs were also developed [4, 37].

Face recognition is one of the major topics in visual recognition [5, 6]. Based on the success of deep learning, many CNNs have been introduced in face recognition and achieved a series of breakthroughs. Similar to image recognition, effective CNNs require a larger amount of training images and larger network size. Yaniv *et al.* [27] trained the DeepFace system with a standard 8 layer CNN using 4.4 million labeled face images. Sun *et al.* [24, 22, 23] developed the Deep-ID systems with more elaborate network architectures and fewer training face images, which achieved better performance. FaceNet [19] was introduced with 22 layers based on the Inception network [25, 33]. It was trained on 200 million face images and achieved further improvement. Parkhi *et al.* [15] introduced the VGG-Face network with up to 19 layers adapted from [21], which was trained by 2.6 million images. This network also achieved comparable results and has been extended to other applications. To overcome the massive request of labeled training data, Masi *et al.* [13] proposed using domain specific data augmentation, which generates synthesis images for CASIA WebFace collection [32] based on differ-

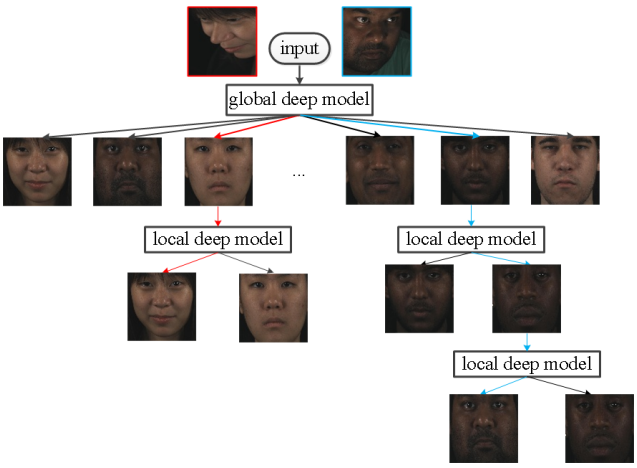


Figure 1: An example of the proposed LCC-CNN framework with a combination of global model and local models. The prediction paths of the two testing face images are indicated in red and cyan.

ent facial appearance variations. Their results trained with ResNet match the state-of-the-art results reported by networks trained on millions of images.

Overall, the previous deep face networks have two limitations. First, data size: training a larger size global model requires more training data, which can be costly and not applicable in certain applications. Second, local information: one deep neural network built over all the class labels may ignore the pairwise local correlations between different labels, which can be used to improve overall performance.

By reconsidering the problem, this paper attempts to improve face recognition performance without changing the architecture of the CNN network. We propose a hierarchical framework that builds chains of local binary CNN classifiers after the global CNN classifier over all the class labels. Hereafter, we refer to these two types of classifiers as local model and global model. The motivation behind this is that a global model focuses more on the global discriminative features over all the classes and tends to misclassify samples from similar classes. With fewer classes, a local model can exploit more local discriminative features for the related labels and can be used to correct the prediction of the global model. With the same training data and network architecture, each local model converges fast and achieves better accuracy than the global model. Also, local models can be trained in parallel, which avoids excessive increase in computational complexity. In addition, when data size is limited, a local model can explore more pairwise label correlations than the global model. To limit the complexity of our framework, we only build binary local models in this paper. Take the UHDB31 face recognition dataset [30] for example. Figure 1 depicts the intuition of the proposed framework. As we can observe, for the first female face image, one binary local model is used to correct the mistake of the global model. Importantly, we can build local models one after another, which leads to a chain of local models to boost performance and avoid error propagation. For the second male face image, a chain of two local models is built to improve the prediction of the global model.

The contributions of this paper include: First, we introduce a hierarchical framework of deep neural networks, where a global model and local models are combined to improve prediction performance. Second, we propose two different ways to create local model chains, which adaptively select a small set of label pairs to build local models. Third, the pairwise correlations between different labels are learned based on their relationships in the score matrices. These correlations are not well explored in global models.

The rest of this paper is organized as follows: in Section 2 we discuss related work. Section 3 describes the proposed framework. The experimental design, results, and analysis are presented in Section 4. Section 5 concludes the paper.

2. Related work

Hierarchical ideas have been introduced to deep neural networks in previous works [28]. Deng *et al.* [3] replaced the flat soft-max classification layer with a probabilistic graphical model that embeds given relationships between labels. Yan [31] proposed a hierarchical deep CNN, where easily distinguished classes are predicted in higher layers while visually similar classes are predicted in lower layers. Murdock *et al.* [14] introduced blockout layers to learn the cluster membership of each output node from fully connected layer. The weight matrix of a blockout layer can be learned from back propagation. These methods modify the global neural network to learn feature representation by embedding clustering information. Local information is introduced either by multi-task learning or weight matrix restriction. However, they still rely on one global model to make predictions for all the class labels. The pairwise label correlations are not explored separately.

The proposed framework is mainly inspired by the Hierarchical Multi-label Classification (HMC) problem, where each sample has more than one label and all these labels are organized hierarchically in a tree or Direct Acyclic Graph (DAG) [20, 1]. Hierarchical information in tree and DAG structures is used to improve classification performance [29, 36, 35, 34]. In visual recognition, each sample only has one label. If we build a meta class label hierarchy, and consider it as an HMC problem, the prediction error of high level nodes will propagate to the prediction of low level nodes. We can also build a local model for each node separately and combine the predictions of all the local models. But this leads to a heavy computation burden, depending on the size of the hierarchy. Thus, in this paper, we pro-

pose building chains of local models after the global model, which possesses the merits of both global and local models.

The proposed framework also inherits features from works about classifier chains for multi-label classification [16, 17]. Similarly, in our framework, we also use local classifier models to exploit the label correlations and select the next layer of local models based on the current prediction. The major difference lies in the output of a local classifier. In our framework, we apply local classifiers to update a single prediction label, rather than leveraging the output of multiple labels.

3. Hierarchical framework with LCC-CNN

The proposed model consists of three major steps: First, select label pairs to build local models based on training data and validation data; two different methods based on similarity matrix and confusion matrix are introduced. Second, build a local classifier for each label pair. Last, combine the prediction results from both the global model and local models to generate a final prediction.

Given a face recognition training dataset $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N}$, x_i and y_i represent the i^{th} sample and the corresponding class label, respectively. The label set is denoted by $\mathcal{C} = \{c_1, c_2, ..., c_l\}$, and we have $y_i \in \mathcal{C}$. First we define the global model as $g(x_i)$, and the global prediction vector of each sample $V_i = \{v_{i1}, v_{i2}, ..., v_{il}\}$ as the last layer output of the global model with size of $1 \times l$. Each value v_{ij} represents the probability of assigning label c_j to the i^{th} sample. From V_i, we can obtain the global prediction result h_i easily, and $h_i \in \mathcal{C}$. All the global prediction results of the dataset are represented by $H = \{h_1, h_2, ..., h_N\}$, thus,

$$V_i = g(x_i), \tag{1}$$

$$h_i = argmax_i(V_i). \tag{2}$$

The goal of our framework is to improve the global prediction results with local models built on a set of label pairs. Let $\mathcal{P} = \{p_{ij}\}$ and $\mathcal{F} = \{f_{ij}(x)\}$ represent the label pair set and the local binary model set, respectively, where p_{ij} and $f_{ij}(x)$ represent the label pair and the local model between label c_i and label c_j, respectively.

3.1. Similarity matrix based local model

First, we consider building local models between visually less discriminative labels. The assumption behind this is that a global model is more likely to confuse between visually similar labels than visually discriminative labels. Take Figure 1 for example: an Asian face image is more likely to be mis-labeled as another Asian face than a non-Asian face. Given a global model, we use the global prediction vector of each sample in validation data as its visual feature vector. Based on this feature map, we can compute the averaged feature vector of each label. Then, the distance between the averaged feature vectors of two different labels can be used to represent their label similarity. Let $\hat{\mathcal{S}} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{\hat{N}}$ represent the validation set, and we can compute the visual feature vector of each sample from the global model as $\hat{V}_i = g(\hat{x}_i)$. We use $\hat{U}_i = \{\hat{u}_{i1}, \hat{u}_{i2}, ..., \hat{u}_{il}\}$ to represent the mean sample feature vector of the i^{th} label c_i, where each value $\hat{u}_{ik}$ represents its k^{th} feature. Then we can define a similarity matrix based on the distance between each pair of mean samples, represented by $W = \{w_{ij}\}^{l \times l}$, where w_{ij} represents the Euclidean distance between the mean samples of c_i and c_j. We have,

$$w_{ij} = \frac{1}{l} \sum_{k=1}^{l} (\hat{U}_{ik} - \hat{U}_{jk})^2. \tag{3}$$

To convert the elements in W to a unique scale, we normalize each element in W by dividing the maximum element of W. Then, each value is subtracted from 1 to represent the similarity instead of distance. Let $Q = \{q_{ij}\}^{l \times l}$ represent the normalized similarity matrix, so that,

$$q_{ij} = 1 - \frac{w_{i,j}}{\max(W)}. \tag{4}$$

For each label, we can find the most similar labels based on the score matrix and a pre-defined threshold t_s. If q_{ij} is larger than t_s, we add a label pair $p_{ij} = \{c_i, c_j\}$ to the label pair set $\mathcal{P}$. This way, we can collect different sizes of label pair sets based on different values of t_s. The pseudo-code of LCC-CNN trained with similarity matrix is summarized in Algorithm 1.

3.2. Confusion matrix based local model

Similarity correlation is not a direct way to measure the prediction performance of a global model. In this subsection, we introduce to use the confusion matrix of the validation set to create the label pair set. Based on the label set $\hat{Y}$ and the global prediction set $\hat{H}$ of the validation set, we can compute the confusion matrix easily, represented by $Z = \{z_{ij}\}$, where $z_{i,j}$ represents the number of samples that have a true label of c_i with $\hat{y} = c_i$ and a prediction label of c_j with $\hat{h} = c_j$. The confusion matrix directly gives pairwise information about the performance of the global model, which matches the motivation of our framework naturally. To obtain a unique scale, we normalize each element in Z by computing the ratio of z_{ij} to the number of samples with true label of c_i; all the normalized values are defined as a matrix $R = \{r_{ij}\}$, where

$$r_{ij} = \frac{z_{ij}}{\sum_1^l (z_{ij})}. \tag{5}$$

If r_{ij} is larger than a pre-defined threshold t_c, we presume that many samples with a true label of c_i are predicted

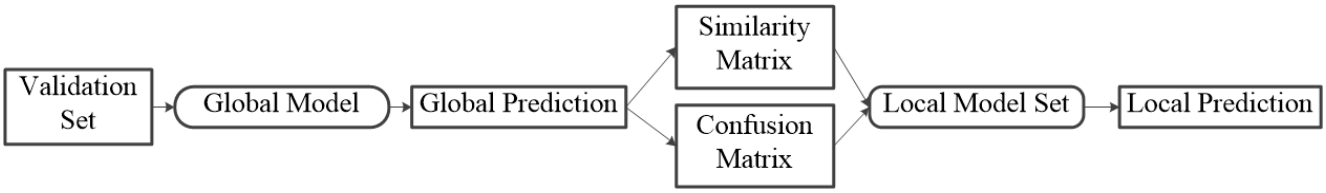


Figure 2: The work-flow of validation-based local model generation.

Algorithm 1: LCC-CNN: Training with similarity matrix

Input: Training set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N}$, validation set $\hat{\mathcal{S}} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{\hat{N}}$ and threshold t_s
Output: global model $g(x)$, label pair set $\mathcal{P}$ and local binary model set $\mathcal{F} = \{f_{ij}(x)\}$

Train global model $g(x)$ with $\mathcal{S}$
Compute global prediction vector of each sample in validation set $\hat{V}_i = g(\hat{x}_i)$ by (1)
Compute mean sample vectors $\hat{U}_i$
Compute similarity matrix W by (3)
Compute normalized similarity matrix Q by (4)
for $i \leftarrow 1$ ***to*** l **do**
  for $j \leftarrow 1$ ***to*** l **do**
    if $q_{ij} > t_s$ **then**
      $\mathcal{P} = \mathcal{P} \cup \{p_{ij}\}$
      train local model $f_{ij}(x)$
      $\mathcal{F} = \mathcal{F} \cup \{f_{ij}(x)\}$
return $g(x)$, $\mathcal{P}$, $\mathcal{F}$;

Algorithm 2: LCC-CNN: Training with confusion matrix

Input: Training set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N}$, validation set $\hat{\mathcal{S}} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{\hat{N}}$ and threshold set t_c
Output: global model $g(x)$, label pair set $\mathcal{P}$ and local binary model set $\mathcal{F} = \{f_{ij}(x)\}$

Train global model $g(X_i)$ with $\mathcal{S}$
Compute global prediction vector of each sample in validation set $\hat{V}_i = g(\hat{x}_i)$ by (1)
Compute global prediction result of each sample $\hat{h}_i = argmax_i(\hat{V}_i)$ by (2)
Compute confusion matrix of validation set $Z = \{z_{ij}\}$
Compute normalized confusion matrix $R = \{r_{ij}\}$ by (5)
for $i \leftarrow 1$ ***to*** l **do**
  for $j \leftarrow 1$ ***to*** l **do**
    if $r_{ij} > t_k$ **then**
      $\mathcal{P} = \mathcal{P} \cup \{p_{ij}\}$
      train local model $f_{ij}(x)$
      $\mathcal{F} = \mathcal{F} \cup \{f_{ij}(x)\}$
return $g(x)$, $\mathcal{P}$, $\mathcal{F}$;

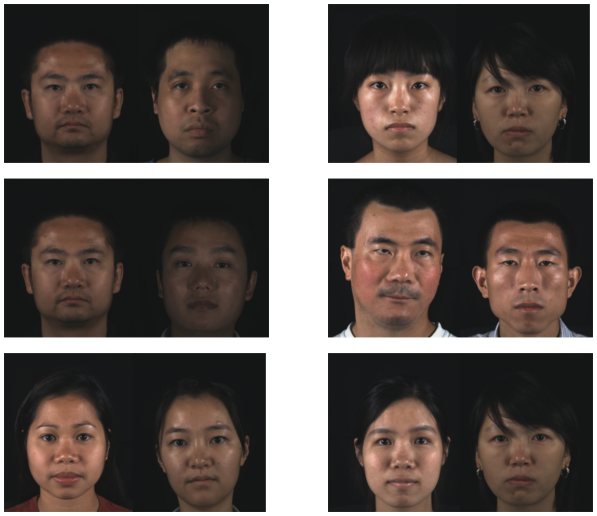
as c_j. Labels c_i and c_j are therefore ambiguous, and a local model is required for further evaluation. To make a robust label pair set and overcome over-fitting, we ignore the direction information in each label pair, so $p_{ij} = p_{ji}$. The pseudo-code of LCC-CNN trained with confusion matrix is summarized in Algorithm 2.

The procedure of building a label pair set in our framework is depicted in Figure 2. Figure 3 shows the top-3 label pairs with highest similarity and confusion scores extracted from the UHDB31 dataset based on the VGG-Face network. Note that if there are many label pairs sharing the same label, we can also consider building a multi-class local model, instead of a set of binary local models. To limit the model complexity, we build chains of binary local models and use the prediction result of one local model to direct the selection of next local model.

3.3. Local classifier chain prediction

Based on the prediction results of both global and local models, we use a top-down strategy to get the final prediction for each testing sample. Starting from the global prediction, each sample will go through a chain of local models to obtain the final prediction, until none of the related local models gives a better prediction. Initially, we set the current prediction label o based on the global prediction result. Then, each model link is built based on two steps. First, all the label pairs containing the current label o in label pair set $\mathcal{P}$ are added to a current matching label set $\mathcal{P}'$. Second, the matching label with the largest prediction result is selected as the next current label. The pseudo-code of LCC-CNN in testing is summarized in Algorithm 3.

In fact, if there is only one matching label and the corresponding local model gives a higher prediction value to this matching label over the current label, we believe this testing sample is mis-classified. Therefore, the current label will be updated to the matching label. If there are multiple matching labels, we need to find the best possible next current label by comparing their prediction values. In this



(a) (b)

Figure 3: The top-3 label pairs with highest similarity and confusion scores on the UHDB31 dataset. (a) From similarity matrix, the scores from top to bottom are 0.716, 0.712 and 0.703. (b) From confusion matrix, the scores from top to bottom are 0.302, 0.286 and 0.238. We can see that each label pair shares some visual similarities that are misclassified by the global model.

process, the next current label does not necessarily have to be the correct label. The local model chain is designed to overcome error propagation in the top-down prediction. If one local model in the chain gives a wrong prediction, the following local models still have chances to correct the error as long as the related label pairs have been added to the label pair set.

4. Experiments

In this section, we present our evaluation of the proposed framework on two types of face recognition scenarios: constrained environment and unconstrained environment. The datasets we use for testing are the UHDB31 dataset [30] and the CASIA-WebFace dataset [32], respectively. We choose the pre-trained VGG-Face model [15] and the pre-trained ResNet-Face model [32] as baseline methods to fine-tune our global model and local models. Since the pre-trained ResNet-Face model is trained on CASIA-WebFace, we use the weights of the pre-trained ResNet101 model from ImageNet [9] on the CASIA-WebFace dataset. The batch size is set to 24 for VGG-Face and 12 for ResNet-Face to limit GPU memory usage. The learning rate is set to 0.001. We assume that the global model converges when the accuracy of the validation set is stable and begins to drop. The local models are built with the samples of the two corresponding labels from the training set. These local models usually converge faster than the global model, and we use the weights at 3,000 iterations during prediction. To observe the overall performance change, we test our LCC-CNN framework based on Similarity Matrix (LCC-CNN-SM) and Confusion Matrix (LCC-CNN-CM) using different threshold sets and different maximum iteration stages. The parameter settings are selected based on the performance on the validation set. All the experiments are conducted on Caffe [10].

Algorithm 3: LCC-CNN: Testing

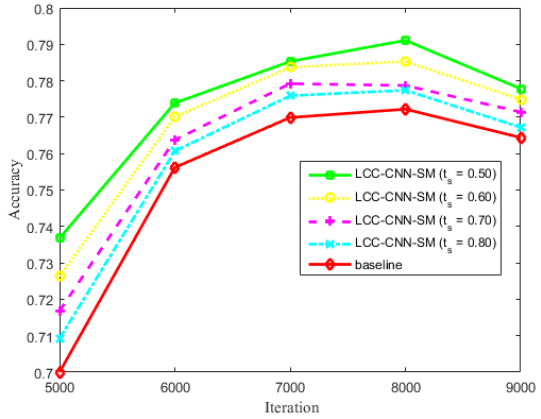
Input: Testing set $\bar{\mathcal{S}} = \{(\bar{x}_i, \bar{y}_i)\}_{i=1}^{\bar{N}}$, global model $g(x)$, label pair set $\mathcal{P} = \{p_{ij}\}$ and local binary model set $\mathcal{F} = \{f_{ij}(x)\}$

Output: Final prediction of testing set $\bar{O} = \{\bar{o}_1, \bar{o}_2, ..., \bar{o}_{\bar{N}}\}$

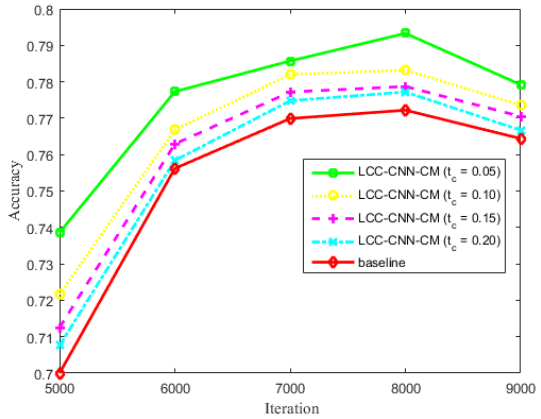
```
for i ← 1 to N̄ do
    Compute global prediction result h̄_i
    Set current prediction label o = h̄_i
    while True do
        Set current matching label set P' = ∅
        Update P', when p_od ∈ P
        if P'! = ∅ then
            Set current predication value a = 0
            Set next current prediction label o_n = o
            for d in P' do
                if argmax(f_od(x̄_i))! = o and
                max(f_od(x̄_i)) > a then
                    o_n = argmax(f_od(x̄_i))
                    a = max(f_od(x̄_i))
            if o_n == o then
                break
            else
                o = o_n
        else
            break
    Set ō_i = o
return Ō;
```

4.1. Constrained face recognition

The UHDB31 dataset [30] contains 29,106 color face images of 77 subjects with 21 poses and 18 illuminations. To evaluate the performance of cross pose face recognition, we use images from 7 close frontal poses for training. The images from the remaining 14 poses are split equally for evaluation and testing. In the UHDB31 dataset, we test our LCC-CNN-SM and LCC-CNN-CM using different threshold sets ($t_s = \{0.50, 0.60, 0.70, 0.80\}$ and $t_c = \{0.05, 0.10, 0.15, 0.20\}$) with global models at different iteration stages from 4,000 to 8,000 for both base-



(a)



(b)

Figure 4: The performance of VGG-Face on the UHDB31 dataset. (a) LCC-CNN-SM. (b) LCC-CNN-CM.

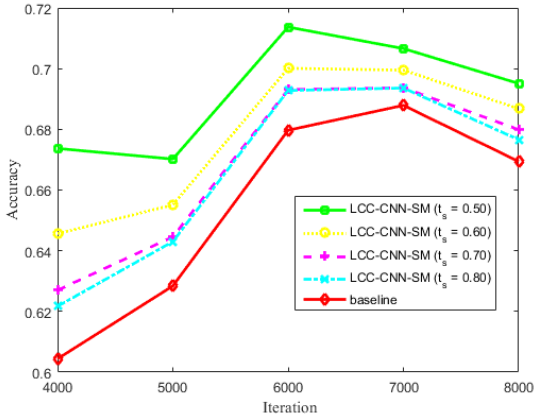
Iterations	SM-0.50	SM-0.60	SM-0.70	SM-0.80	CM-0.05	CM-0.10	CM-0.15	CM-0.20
5,000	105	62	23	10	127	47	25	6
6,000	75	49	21	9	95	36	17	6
7,000	70	47	20	8	76	36	13	6
8,000	71	41	17	8	88	36	14	6
9,000	70	47	18	7	88	38	14	4

Table 1: The sizes of local model sets of VGG-Face on the UHDB31 dataset.

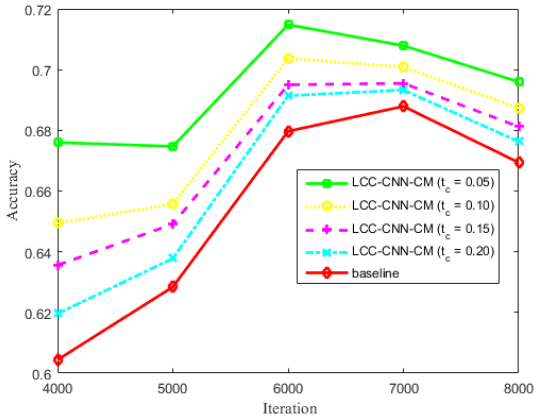
Iterations	SM-0.50	SM-0.60	SM-0.70	SM-0.80	CM-0.05	CM-0.10	CM-0.15	CM-0.20
5,000	151	55	19	10	177	66	28	11
6,000	134	48	19	14	160	56	28	11
7,000	96	32	11	9	117	39	13	6
8,000	117	47	9	8	139	53	16	7
9,000	132	59	19	14	153	66	28	10

Table 2: The sizes of local model sets of ResNet-Face on the UHDB31 dataset.

lines. The results are shown in Figures 4-5 and Tables 1-2.



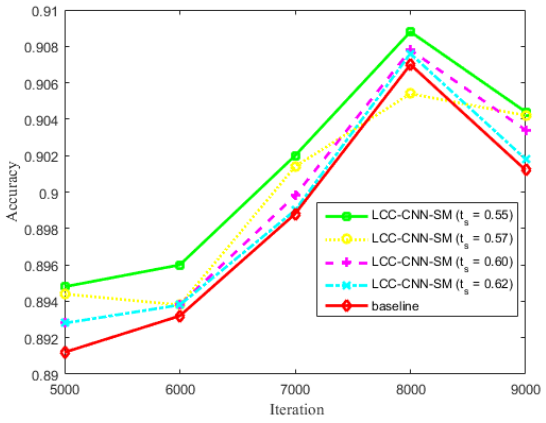
(a)



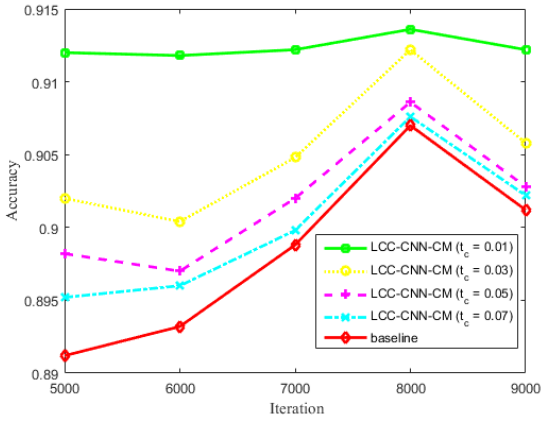
(b)

Figure 5: The performance of ResNet-Face on the UHDB31 dataset. (a) LCC-CNN-SM. (b) LCC-CNN-CM.

From Figure 4, we can observe that the VGG-Face global model achieves the best performance (77.2%) at 8,000 iterations. Both LCC-CNN-SM and LCC-CNN-CM improve the performance at different training stages. LCC-CNN-SM and LCC-CNN-CM achieve the best performance of 79.1% and 79.3% when $t_s = 0.50$ and $t_c = 0.05$, respectively. From Figure 5, we can observe that the best performance of the ResNet-Face model (68.8%) is lower than that of the VGG-Face model. Both our models perform better than the baseline methods at different training stages. The best results of LCC-CNN-SM (71.3%) and LCC-CNN-CM (71.5%) are also achieved when $t_s = 0.50$ and $t_c = 0.05$, respectively. Overall, the performance is limited by the challenge of cross-pose evaluation. From Tables 1-2, we can observe that as t_s and t_c decrease, more label pairs are selected at different training stages.

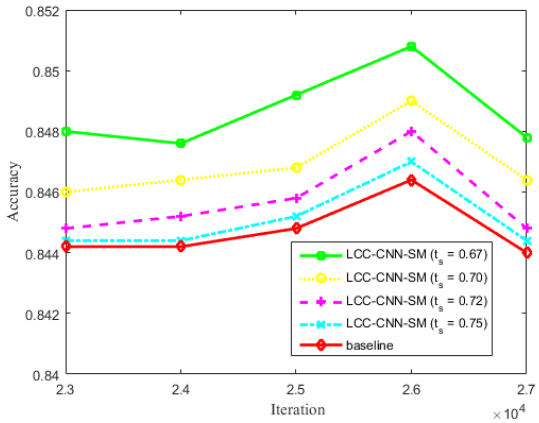


(a)

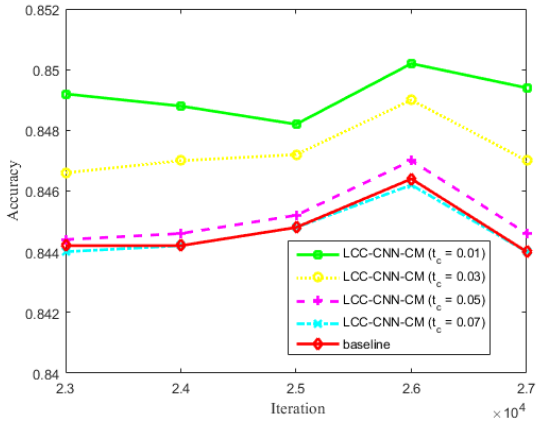


(b)

Figure 6: The performance of VGG-Face on the CASIA-WebFace dataset. (a) LCC-CNN-SM. (b) LCC-CNN-CM.



(a)



(b)

Figure 7: The performance of ResNet-Face on the CASIA-WebFace dataset. (a) LCC-CNN-SM. (b) LCC-CNN-CM.

4.2. Unconstrained face recognition

The CASIA-WebFace dataset [32] contains 494,414 face images in the wild from 10,575 subjects. We randomly select 100 subjects that each contain more than 100 images to build a face identification subset. Then, the subset is divided into a training set, a validation set and a testing set, which contain 50%, 25% and 25% of the images per subject, respectively. In the CASIA-WebFace dataset, we test our LCC-CNN-SM and LCC-CNN-CM using different threshold sets ($t_s = \{0.55, 0.57, 0.60, 0.62\}$ and $t_c = \{0.01, 0.03, 0.05, 0.07\}$) with the VGG-Face model at different iteration stages from 5,000 to 9,000. We test our LCC-CNN-SM and LCC-CNN-CM using different threshold sets ($t_s = \{0.67, 0.70, 0.72, 0.75\}$ and $t_c = \{0.01, 0.03, 0.05, 0.07\}$) with the ResNet-Face model at different iteration stages from 23,000 to 27,000. The results are shown in Figures 6-7 and Tables 3-4.

Iterations	SM-0.55	SM-0.57	SM-0.60	SM-0.62	CM-0.01	CM-0.03	CM-0.05	CM-0.07
5,000	175	108	43	14	391	60	17	4
6,000	171	94	35	17	418	62	19	8
7,000	155	87	22	8	372	63	13	4
8,000	173	108	46	23	383	48	12	4
9,000	335	198	88	46	416	57	17	10

Table 3: The sizes of local model sets of VGG-Face on the CASIA-WebFace dataset.

Iterations	SM-0.67	SM-0.70	SM-0.72	SM-0.75	CM-0.01	CM-0.03	CM-0.05	CM-0.07
23,000	229	107	59	21	601	78	12	2
24,000	218	103	57	19	605	82	11	1
25,000	235	108	61	22	605	82	14	3
26,000	232	109	61	20	593	82	15	4
27,000	230	107	59	20	598	87	14	1

Table 4: The sizes of local model sets of ResNet-Face on the CASIA-WebFace dataset.

From Figure 6 and Figure 7, we can observe similar im-

provements on both baselines. From Figure 6, we can observe that the VGG-Face global model achieves the best performance (90.7%) at 8,000 iterations. Both LCC-CNN-SM and LCC-CNN-CM improve the performance at different training stages. LCC-CNN-SM and LCC-CNN-CM achieve the best performance of 90.9% and 91.4% when $t_s = 0.55$ and $t_c = 0.01$, respectively. From Figure 7, we can observe that the best performance of the ResNet-Face model (84.6%) is lower than that of the VGG-Face model. This is because the pre-trained model is trained on ImageNet rather than a face dataset. Both our models perform better than the baseline at different training stages. The best results of LCC-CNN-SM (84.9%) and LCC-CNN-CM (85.0%) are achieved when $t_s = 0.67$ and $t_c = 0.01$, respectively. From Tables 3-4, we can observe that as t_s and t_c decrease, more label pairs are selected at different training stages.

Figure 8 depicts several prediction examples of our framework. From these examples, we can observe that the global model fails to distinguish visually similar faces, especially when there are similar facial attributes (such as "bald head" in (b), "black skin" in (c) and "wavy hair" in (d)). On the other hand, local models explore more pairwise correlations between labels and can extract more locally discriminative features, especially for the related labels. Also, our framework can explore chains of similar labels like the faces in (c) and (f). This information can help us understand the pairwise correlations between different faces.

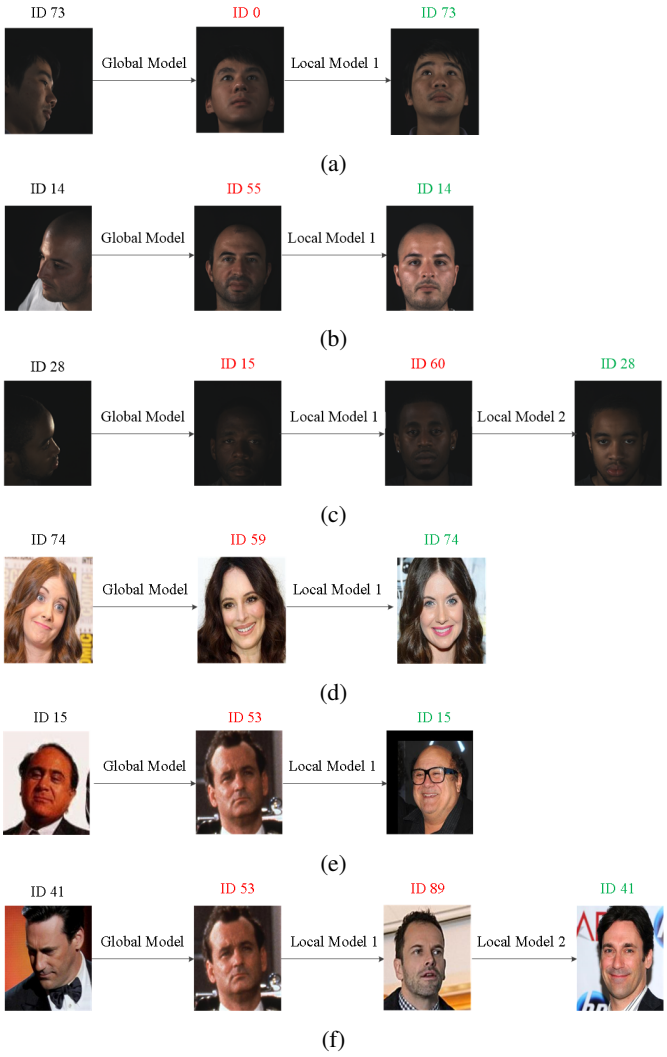


Figure 8: The prediction examples of our hierarchical framework. The text on the top of each image represents its label, where black, red and green indicate ground truth label, mistaken label and correct label, respectively.

5. Conclusion

In this paper, we propose a hierarchical face recognition framework that combines the merits of both global and local models. Chains of local models built based on a similarity matrix and confusion matrix are used to verify the prediction of a global model. The accuracy is improved by 2.7% and 0.7% on the UHDB31 dataset and the CASIA-WebFace dataset, respectively. However, there are still several open questions about the current framework that we need to address in the future. First, as the number of faces increases, the local model selection during testing becomes more crucial and requires more attention. Currently, we rely on local prediction results to select the next local model. The local pair sets must be sufficient to overcome the error propagation problem. Second, there is a tradeoff between the performance of the global model and the final prediction. In the future, we will investigate whether it is possible to build local models based on an unconverged early stage global model, but still achieve comparable performance to the current framework. This way we can find a better balance between global and local models.

Acknowledgements

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2015-ST-061-BSH001. This grant is awarded to the Borders, Trade, and Immigration (BTI) Institute: A DHS Center of Excellence led by the University of Houston, and includes support for the project "Image and Video Person Identification in an Operational Environment: Phase I" awarded to the University of Houston. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

References

[1] W. Bi and J. Kwok. Mandatory leaf node prediction in hierarchical multilabel classification. *IEEE Transactions on Neural Networks and Learning Systems*, 25(12):2275–2287, 2014.

[2] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng. Large scale distributed deep networks. In *Proc. Neural Information Processing Systems*, pages 1223–1231, Lake Tahoe, NV, Dec. 3-8 2012.

[3] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *Proc. European Conference on Computer Vision*, pages 48–64, Zrich, Switzerland, Sept. 6-12 2014.

[4] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Proc. Advances in Neural Information Processing Systems*, pages 1269–1277, Montreal, Canada, Dec. 8-13 2014.

[5] C. Ding, J. Choi, D. Tao, and L. S. Davis. Multi-directional multi-level dual-cross patterns for robust face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):518–531, 2016.

[6] P. Dou, L. Zhang, Y. Wu, S. K. Shah, and I. A. Kakadiaris. Pose-robust face signature for multi-view face recognition. In *Proc. Biometrics Theory, Applications and Systems*, pages 1–8, Arlington, VA, Sept. 8-11 2015.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, 2016.

[8] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *Proc. International Conference on Learning Representations*, San Juan, Puerto Rico, May 2-4 2016.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. Computer Vision and Pattern Recognition*, Las Vegas, NV, June 26 - July 1 2016.

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Neural Information Processing Systems*, pages 1097–1105, Lake Tahoe, NV, Dec. 3-8 2012.

[12] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[13] I. Masi, A. Tran, T. Hassner, J. T. Leksut, and G. Medioni. Do we really need to collect millions of faces for effective face recognition? In *Proc. European Conference on Computer Vision*, Amsterdam, Netherlands, Oct. 8-16 2016.

[14] C. Murdock, Z. Li, H. Zhou, and T. Duerig. Blockout: Dynamic model selection for hierarchical deep networks. In *Proc. Computer Vision and Pattern Recognition*, Las Vegas, NV, June 26 - July 1 2016.

[15] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proc. British Machine Vision Conference*, volume 1, page 6, Swansea, UK, Sept. 7-10 2015.

[16] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. In *Proc. Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 254–269, Bled, Slovenia, Sept. 7-11 2009. Springer.

[17] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359, 2011.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[19] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. Computer Vision and Pattern Recognition*, pages 815–823, Boston, MA, June 8-10 2015.

[20] C. N. Silla Jr and A. A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72, 2011.

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. International Conference on Learning Representations*, San Diego, CA, May 7-9 2015.

[22] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Proc. Advances in Neural Information Processing Systems*, pages 1988–1996, Montreal, Canada, Dec. 8-13 2014.

[23] Y. Sun, D. Liang, X. Wang, and X. Tang. DeepID3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015.

[24] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Proc. Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. Computer Vision and Pattern Recognition*, Boston, MA, June 8-10 2015.

[26] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *Proc. Advances in Neural Information Processing Systems*, pages 2553–2561, Lake Tahoe, Nevada, Dec. 5-10 2013.

[27] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the gap to human-level performance in face verification. In *Proc. Computer Vision and Pattern Recognition*, pages 1701–1708, Columbus, OH, June 24-27 2014.

[28] A.-M. Tousch, S. Herbin, and J.-Y. Audibert. Semantic hierarchies for image annotation: A survey. *Pattern Recognition*, 45(1):333–345, 2012.

[29] G. Valentini. True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(3):832–847, 2011.

[30] Y. Wu, S. K. Shah, and I. A. Kakadiaris. Rendering or normalization? an analysis of the 3d-aided pose-invariant face recognition. In *Proc. International Conference on Identity, Security and Behavior Analysis*, pages 1–8, Sendai, Janpan, Feb. 29 - March 2 2016.

[31] Z. Yan, V. Jagadeesh, D. Decoste, W. Di, and R. Piramuthu. HD-CNN: Hierarchical deep convolutional neural network for image classification. In *Proc. International Conference on Computer Vision*, Santiago, Chile, Dec. 13-16 2015.

[32] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[33] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proc. European Conference on Computer Vision*, pages 818–833, Zurich, Switzerland, Sept. 6-12 2014.

[34] L. Zhang, P. Dou, S. K. Shah, and I. A. Kakadiaris. Hierarchical multi-label framework for robust face recognition. In *Proc. International Conference on Biometrics*, pages 127–134, Phuket, Thailand, May 19-22 2015.

[35] L. Zhang, S. Shah, and I. Kakadiaris. Hierarchical multi-label classification using fully associative ensemble learning. *Pattern Recognition*, 70(C):89 – 103, 2017.

[36] L. Zhang, S. K. Shah, and I. A. Kakadiaris. Fully associative ensemble learning for hierarchical multi-label classification. In *Proc. British Machine Vision Conference*, Nottingham, UK, Sept. 1-5 2014.

[37] X. Zhang, J. Zou, K. He, and J. Sun. Accelerating very deep convolutional networks for classification and detection. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):1943–1955, 2016.