Neurocomputing 215 (2016) 232–240

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Cascade shallow CNN structure for face verification and identification

Biao Leng [a], Yu Liu [a], Kai Yu [a], Songting Xu [a], Ziqing Yuan [b], Jingyan Qin [c,*]

[a] School of Computer Science and Engineering, Beihang University, Beijing 100191, PR China
[b] South-Central University for Nationalities, Wuhan 430074, PR China
[c] School of Mechanical Engineering, University of Science & Technology Beijing, Beijing 100191, PR China

ARTICLE INFO

*Article history:*
Received 19 February 2015
Received in revised form
1 August 2015
Accepted 2 August 2015
Available online 11 June 2016

*Keywords:*
Face recognition
Cascade convolutional network
Local features
Sub-region division

ABSTRACT

Face recognition is a long-standing challenging topic in computer science, especially on insufficient datasets. The obstacle also lies in the balance of speed and accuracy. Recently, many algorithms claim that they have obtained great performance with high accuracy, but they are not enough for real-time application. In this work, a novel fast and accurate solution is proposed to deal with the face recognition problem on the small training set. Based on face alignment, we present two methods to extract features. One is a combination of several kinds of human designed feature descriptors applied on patches partitioned according to facial landmarks. The other one is a cascade classifier based on shallow convolutional neural networks. Both methods can represent the face as a set of feature vectors, which can be dealt with SVM or a boosting verification algorithm in this work. In the experiments, the proposed framework has achieved great performance for face recognition and verification with high speed and high accuracy, based on the public available datasets such as the Labeled Face in the Wild dataset and the AT&T database of faces.

## 1. Introduction

Face recognition is a hot research topic in computer science. Given an image or a sequence of images, the face recognition task is to locate regions (for videos, in each moment) that contain faces, and predict the identities of the faces. It is increasingly used in many real world applications, ranging from surveillance scenarios to entertainment applications. New techniques guaranteeing speed and accuracy are on great demands, and indeed, they are still developing rapidly recent years. Inspiring, some algorithms have reached the accuracy of human performance [1,2], which maybe for the first time widely attracts the public's interest.

Most face recognition works are in essence finding solutions for two tasks: extracting features and using features to classify faces. Many feature extractors have been introduced, and they can be mainly divided into two sorts: human-designed feature extractors and deep-learning feature extractors. The first sort contains many classical feature descriptors like LBP [3], LPQ [4], BSIF [5], and HOG [6]. They can usually run in a high speed, and their discrimination ability has been examined in many works. Also, they can be flexibly integrated to many algorithm frameworks, reaching higher performance without changing their own form. On the other hand, the deep-learning feature extractors are usually the lower layers of deep learning structures. Due to the full connections or convolution operations, these extractors are usually computationally expensive. However, they can discover deep features, which are hard for human-designed descriptors to realize. Their power is also shown in many recent works.

Likewise, the obstacle of face recognition always lies in the balance of speed and accuracy. For example, though highly accurate, algorithms using conventional deep neural networks cannot reach a satisfying speed on normal computing devices without high ability of parallel computation, thus unable to fit in some real-time applications. And another problem is, in the real world, few samples of a single subject can be collected, which affects the training of deep structures as well as many other algorithms such as SVMs and Softmax, because all these algorithms address the problem of different face orientation, expression and illumination by feeding abundant samples into the training procedure. With insufficient samples, these algorithms can only acquire information in limited conditions, thus losing generality.

To address this problem, there are efforts trying to apply face alignment to the recognizing procedure. Some alignment algorithms like congealing [7] and 3D deep-face model alignment [1] aim to turn the tilted faces into frontal ones. Another kind of alignment algorithms do not explicitly perform image transformation, but instead label the facial landmarks such as eye centers and nose tips, and leave the work to the subsequent processes. This kind of approaches can be dated back to the classical active shape model (ASM) [8], and has been developed over the years.

[*] Corresponding author.
*E-mail address:* qinjingyanking@foxmail.com (J. Qin).

http://dx.doi.org/10.1016/j.neucom.2015.08.134
0925-2312/© 2016 Elsevier B.V. All rights reserved.

How to make use of these facial landmarks remains an open task for researchers.

In this work, we manage to solve the issues mentioned above, that is the efficiency problem and the problem of insufficient samples for target subjects. We adopt a highly efficient joint face detection and alignment algorithm, and design two kinds of feature extractions, which are based on human-designed feature descriptors and deep-learning. Both these methods can rapidly utilize the facial landmarks to generate a general model using few samples from target subjects without image transformation. The latter method is shown to surpass the former one, since its cascade of shallow convolutional neural networks (CNNs) can extract more discriminative features and at the same time run efficiently.

The two contributions of this paper are as follows:

- We propose a new method to utilize human-designed feature extractors. Though surpassed by another method of our own, the feasibility of this method might bring us some inspiration.
- We design a boosting cascade of shallow CNNs for feature extraction. This framework features both high efficiency and robustness. It can be pre-trained by large extra datasets, so does not require much samples from target subjects. Furthermore, the extracted features, named as ShallowIDs, are generally adequate to be applied to both verification and identification tasks. Once trained, the model can be modified flexibly.

This paper is organized as follows. In Section 2, we give a brief review of some related works on face recognition. We introduce the detailed implementation of our algorithm in both Sections 3 and 4. We show results of experiments conducted on two public benchmark datasets, the Labeled Face in the Wild dataset (LFW) [9] and the AT&T database of faces (formerly the ORL database) [10], to verify our algorithms. A brief conclusion will be made in Section 6.

## 2. Related work

In the field of computer vision, many techniques have been developed for different aspects, including image processing, image understanding and applications based on pattern recognition. For image processing and understanding, Liu et al. introduce a solution for multiview depth video coding problem [11]. Yang et al. present an efficient way for image quality prediction [12] and also a bundled-optimization model for dynamic scene reconstruction [13]. As for applications, research topics can be further divided into object retrieval [14–16], image search [17], face recognition [18], etc. With the powerful representation ability, deep learning is widely studied and successfully applied in many fields, such as image processing [19], speech recognition [20], natural language processing [21] and 3D model retrieval [22–24].

In recent years, face recognition has become a research hotspot. The accuracy of the best face verification algorithms has almost reached human level. Works of face verification can be generally classified into two categories: some focus on developing traditional statistics-based methods, and the others make efforts to design better deep-learning structures. Traditional statistics-based methods [25–27] usually use human-designed feature extractors, and try to improve performance by modifying the algorithm framework or some detailed process. And just like the extractors themselves, these methods can run in high speed but cannot reach satisfying accuracy.

On the contrary, the power of deep structures has been shown in [1,18,28–30]. These works build convolution neural networks consisting of several layers and manage to find out the deep features laid in the face images. To train the deep networks, usually extremely large training sets are needed to guarantee convergence as well as to avoid over-fitting. Besides, careful training steps and meticulous tricks are also essential for training [31]. Among these works, Taigman et al. propose a method using a 3D model to align faces, achieving high accuracy [1]. In their work, facial landmarks are first located on the image by a 2D alignment algorithm, together with affine transformation turning the face to be a bit more frontal. Then regions fine split referring to the facial landmarks are mapped onto a 3D model. In this way, image of the face from any perspective can be generated by placing a virtual camera at a specific position.

As for 2D facial landmark locating, the most classical algorithms are the Active Shape Model (ASM) algorithm [8] and the Active Appearance Model (AAM) algorithm [32]. After them, many works try to develop these two models in different directions. An algorithm called explicit shape regression (ESR) [33] shows high speed together with high accuracy, but is recently surpassed by an even faster algorithm proposed in [34]. Then an algorithm proposed creates a framework to jointly do face detection and alignment, making use of these landmarks to enhance detection and reducing the time cost of the whole procedure [35]. Also some algorithms make efforts to solve alignment problems under occlusion [36], using models similar to DPM [37].

In the case of insufficient training samples, there are also researches trying to find a robust solution. Some of them focus on an extreme condition where only one sample from each subject is given for training [38–40]. The common strategy is to precisely locate and extract local features. To address this problem, some algorithms extract local rectangle patches [39] and others try to reconstruct the 3D face model [40]. And some algorithms turn to conduct virtual sample extension on the given datasets. Erhu et al. propose a virtual image generating method based on local feature extraction [41]. The 3D model proposed in [1] can also be used for sample augmentation, though the authors of it use it only for frontalization.

Another important sub-procedure of face recognition is the face extraction. As mentioned before, it can be divided into Deep-learning methods and human-designed methods. Deep-learning feature extraction method can be referred in [1]. Not surprisingly, it is followed by a very deep convolutional neural network in that work. Contrast to rather monotonous design of deep-learning feature extractors, there are many kinds of human-designed descriptors. The algorithms LBP [3], LPQ [4] and BSIF [5] are regarded to be three of the best face descriptors up to now. Beside them, the HOG descriptor [6] is also a powerful feature descriptor, but usually applied in object detection.

## 3. Preparations: an attempt using joint human-designed features

Before introducing the cascade shallow CNN algorithm, we describe an idea jointly using three human-designed features: LBP, LPQ and BSIF. This idea is actually a discarded idea in the early stage of our work, but the cascade shallow CNN algorithm is to some extent inspired by it, so we also present it and use it for following comparisons.

### 3.1. Detection and alignment

In recent years, researches on face alignment have made significant progress. Among them, the method called “joint cascade face detection and alignment” [35] is applied to detect faces in images and locate facial landmarks, such as the eyes corners and nose tips. This method shown in experiments is highly precise and attains amazing speed at the same time.

We represent all the locations of facial landmarks as a vector, called the 'shape', which is a concatenation of the coordinates of the points. Suppose that we are going to locate $L$ facial points, then the dimension of a shape vector is $L \times 2$.

The classification and alignment is in a joint cascade manner, divided into T stages. In each stage, there are $N = L \times K$ full decision trees ($K$ is a constant set by the designer), each can be viewed as a weak classifier. Each tree takes the image and a currently estimated shape as input, and outputs a score for the image, indicating the possibility that the image restricted by the shape is a face as well as a binary vector, which contains only one '1', showing which leaf the input has fallen into. A threshold is set in each tree, then an input resulting an accumulated score lower than the threshold is immediately rejected. At the end of each stage, the binary vectors output by all the trees are concatenated into a long vector, which is then multiplied by a global linear regression matrix $W$ to become an estimated offset of the shape. The shape is updated with this offset and becomes the input for the next stage. After this procedure, non-face images are filtered out, and the shape is aligned to represent the real positions of the facial key points.

In each tree, each node tests the input according to pixel difference feature, which is the difference between two pixels located by adding a pair of coordinate offsets stored in the node to the current position of one specified facial point. Note that the referred facial points for all the nodes in one tree are the same. The feature is compared to a threshold stored in the node to decide which child node the input is going to fall in. In the training phase, the selection of the coordinate offsets and thresholds aims to reduce the entropy of the input dataset. Actually, the entropy can be calculated according to classification (faces and non-faces distribution) as well as alignment (shape variance). The strategy is to do more classification tests in early stages, filtering out as many non-face images as possible to assure the speed.

The mentioned matrix $W$ is learnt at the end of each stage by a linear regression minimizing the following objective function:

$$\min_{W^t} \sum_{i=1}^{N} \left\| \Delta \hat{S}_i^{\,t} - W^t \Phi^t (I_i, S_i^{t-1}) \right\|_2^2 + \lambda \, \|W^t\|_2^2 \tag{1}$$

where the first term is the regression target, the second term is a $L_2$ regularization on $W^t$, the transposition of $W$, and $\lambda$ controls the regularization strength. Inside the first term, the $S$ is the ground-truth shape, and the $\Phi$ is the concatenated binary vector generated in the same manner as mentioned above, by the newly trained decision trees in the stage.

We have also tried another strategy, where detection is accomplished by OpenCV, and alignment is done by the method proposed in [34]. The results of experiments show that the former discussed method is much faster and performs much better on profiles.

### 3.2. Feature extraction

Local Binary Patterns (LBP) [3] is a classical and widely applied human-designed feature descriptor. On each pixel, it scans the neighboring pixels and compares them to the central pixel, generating a binary sequence representing the value relationships. Local Phase Quantization (LPQ) [4] is a relatively new descriptor, featuring high insensitivity to centrally symmetric blur. It applies a short-term Fourier transformation (STFT) over a rectangular $M - by - M$ neighborhood $N_x$ at each pixel position $x$ of an image $f(x)$:

$$F(u, x) = \sum_{y \in N_x} f(x - y) e^{-j2\pi u^T y} = w_u^T f_x \tag{2}$$

where $w_u$ is the basis vector of the $2 - D$ DFT at frequency $u$, and $f_x$ is another vector containing all $M^2$ image samples from $N_x$. This procedure can be highly boosted in an implementation using 2-D convolutions:

$$f(x) * e^{-2\pi j u^T x} \tag{3}$$

for all $u$ and considering only four complex coefficients, which correspond with $2 - D$ frequencies $u_1 = [a, 0]^T$, $u_2 = [0, a]^T$, $u_3 = [a, a]^T$, $u_4 = [a, -a]^T$ where $a$ is a scalar frequency. Then it performs whitening transformation and quantization.

Binarized Statistical Image Features (BSIF) [5] is also a local binary descriptor. However, unlike the LBP and LPQ, its filters are learnt from natural images through Independent Component Analysis (ICA). The binary code string $b$ at each window is obtained by binarizing each element $s_i$ of the filtering result as follows:

$$b_i = \begin{cases} 1, & s_i > 0 \\ 0, & otherwith \end{cases}$$

where $b_i$ is the $i$th element of $b$.

These three methods take a rectangular patch of pixels as input and produce a histogram of codes generated on the patch. This enables us to combine these three descriptors by feeding a same input to them and concatenating their outputs into a final feature vector. Also, we do not use the original pixel patches cut out from the image as the input. Instead, we do sampling on shape-specified patches, obtaining matrices of pixels, and then feeding them into the descriptors.

We divide the face image into patches by a pre-defined map consisting of 68 facial landmarks and edges connecting them. These landmarks are located by the above mentioned alignment algorithm, and edges divide the face into several regions. We design the division based on two principles: the surface specified by each region on the real face is as flat as possible, and each region is as large and rectangular as possible. The first principle is based on the fact that the features inside a surface vary jointly and linearly when illumination and orientation changes. The second principle is based on the fact that facial features have stretch ability to a certain degree. Enlarging the regions can guarantee a relatively higher robustness. Assuming that the regions are flat enough and the location of landmarks are precise enough, in this way we can generate a general representation of face with even one single sample by normalizing the features extracting on each region respectively. As stated above, we do sampling on these patches (72 × 72 sampling points located referring the ratio to the edges in our experiments). The partition and sampling operation is shown in Figs. 1 and 2. We apply the local feature descriptors on the sampled data matrices to get a series of histograms. The histograms are concatenated to be the final high dimensional vector.

Besides, the local feature descriptors can be specified to a window size to extract features at different scales. We choose window sizes of 7, 5 and 3 for LBP, LPQ and BSIF respectively because the latter two descriptors' speed is affected by the window size but LBP's is not.

At this stage, the extracted features are collectively called LLB.

## 4. Deep cascade local features

In order to achieve higher accuracy, we need to find more powerful features to represent faces. The LBP and LPQ feature descriptors discussed above are deliberately designed by human, thus might lose some discriminative information, and BSIF is still not satisfying enough. Here we utilize deep learning to extract more powerful features. The power of deep learning has been shown in a bunch of latest works [42,43,19,44]. Furthermore, we

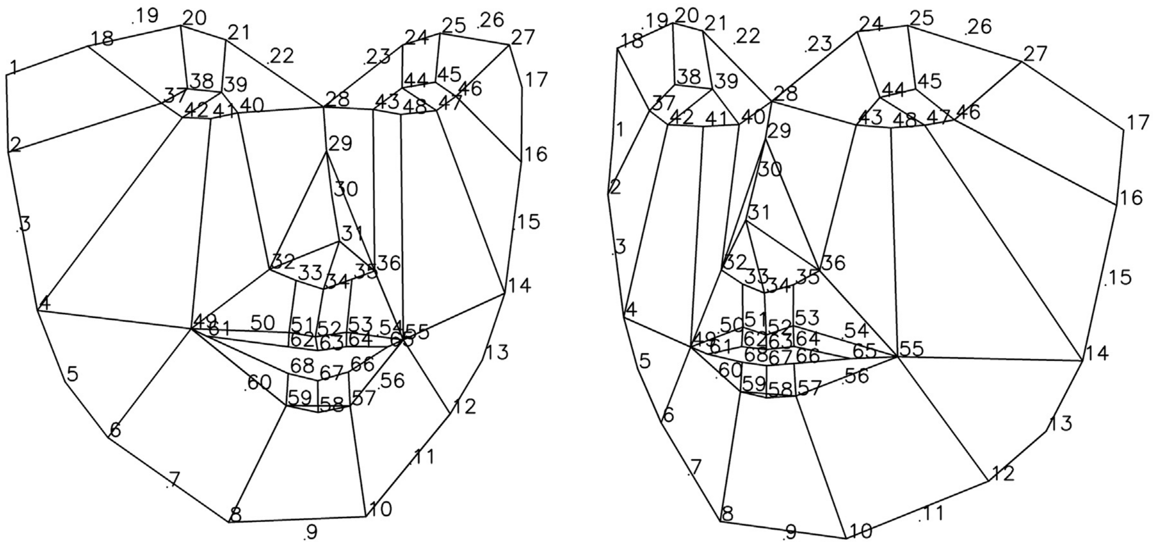


**Fig. 1.** Partitions of the face specified by the facial landmarks. Each sub-region is an irregular quadrangle. The surfaces on the real face are approximately flat, leading to invariance to illumination and gesture changing. And also, the surfaces are designed relatively large due to facial features' flexibility.

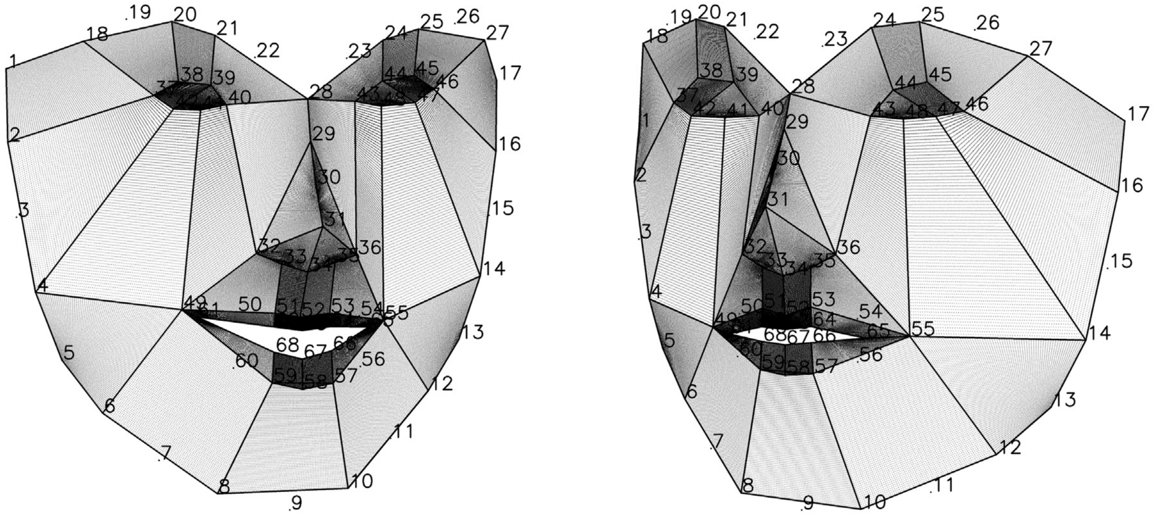


**Fig. 2.** Sampling is done in the sub-regions (72 × 72 pixels in our experiments). The sampled pixels are stored inside matrices, then fed to local feature extractors.

find that information extracted from some certain parts is sufficient extent for identification, so we develop a cascade structure of shallow CNNs.

The boost CNN cascade framework constitutes several shallow CNNs. With output connected to a verification unit, each can be viewed as a weak classifier. Pairs of face images belonging to different subjects would be rejected at any of these CNNs.

### 4.1. Overall framework

In pursuit of fast and precise verification, we design the "$boost - cnn$" structure. We train several shallow convolutional networks as the weak classifier, and employ them to verify faces in cascade. If two faces do not belong to a same person, the former $weak - cnn$ will give the reject signal.

- Firstly, we detect and resize the face area to 224 × 224, and use [34] to get the landmarks, as same as the method used in Section 3.
- We choose the five landmarks (left eye center, right eye center, nose apex, left and right corner of the mouth) as the five regions' centers. Each region patch has 2 scales (40 × 40 pixels and 80 × 80 pixels).
- In training phase, all patches in the same region of different people are used to train the model. Meanwhile the class-id, the personal identity number, serves as the label. After these networks are pre-trained, the final classification layer of CNNs is removed and a cascade structure will be trained for verification whether the local features match the target face's. 10 CNNs with same structure will be trained in this stage. The details are shown in Fig. 3.
- In testing phase, different regions in face will be identified by the corresponding network in serial.

### 4.2. Structure of CNN

We design two convolutional neural network structures ($40CNN$ and $80CNN$) to extract local and half-global features respectively. The $40CNN$ is faster while the $80CNN$ discovers more details.

- $40CNN$: The size of first input layer of 40CNN is set to 40 × 40 × 3. All 40 × 40 local patches at the same location of all the samples (RGB 3-channel face images) are used to train the network. The second layer is a convolution layer with a kernel

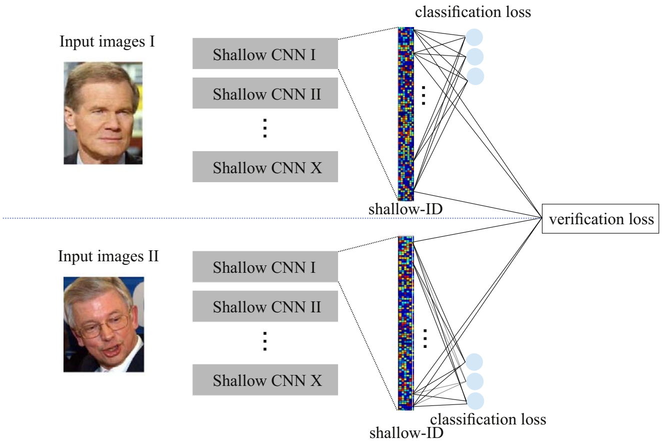


**Fig. 3.** Detail of pre-train and fine-tune strategy. First, we use class-id to supervise the training process. ShallowIDs are followed by softmax classifier layers. After pre-trained, we replace softmax-loss with verification loss (twofold loss) to do the verification training.

size 5 × 5, stride size 2 × 2 and feature number 64, followed by a max-pooling layer $MP1$ with a kernel size 3 × 3 and stride size 2. The third layer is also a convolution layer with a kernel size 3 × 3, stride size 2 × 2 and feature number 256, followed by a max-pooling layer $MP2$ with a kernel size 5 × 5. So the output of the $MP2$ is a 256-dimension vector. It is input to a full connection layer outputting a 1000-dimension vector, called the ShallowID. In classification task, the ShallowID is directly used by a Softmax layer, and in verification task, two ShallowIDs are input to a verification unit. The activation function of these layers is the rectified linear unit (ReLU) [Rectified linear units improve restricted Boltzmann machines].
- $80CNN$: If the confidence probability of the 40CNN cannot reach a preset threshold, we apply the $80CNN$ to do the final decision. It has a similar structure with $40CNN$, the details of $80CNN$ are shown in Fig. 4.

The motivation we design the local shallow CNNs is that in different regions, different set of discriminatory features should be extracted. In this way, we can reduce the kernel number of each CNN and make the feature more robust.

### 4.3. Pre-train and fine tune

Due to the insufficiency of training data in benchmark database, we use extra data to pre-train the CNNs with a learning rate at 0.02. In this stage we consider the softmax-loss as the objective function to perform classification-based training. The coarse CNNs are then fine tuned by using twofold loss with training data in the benchmark database. Learning rate in fine tune stage is set to 0.002. After that, the output of second last layer (full connection layer) will be used to represent faces.

#### 4.3.1. Pre-train with large coarse dataset

In order to pre-train the model with big data, we gather 180,000 images on the Internet, which contain 1200 subjects. The output dimension of the softmax layer is set to 1200, then the class-id acts as the supervised signal. Again, the ShallowID output by the pre-trained network can be used for verification task.

We have an interesting finding that after the 10th iteration of training, the loss calculated on the training dataset remains steady, but the model can be observed to keep improving if tested on the benchmark datasets (see Fig. 5). This means further training can indeed enhance generality.

#### 4.3.2. Fine tune with twofold loss

Because we use ShallowIDs to represent faces for verification tasks, the efficiency of our pipeline is significantly related to the discrimination of ShallowIDs. In pre-train phase, we used classification signal to distinguish person from person. But in this phase, we use verification signal to fine tune the model trained just now.

We define the twofold loss as the loss function:

$$loss_{veri} = label_i \cdot \| conv(I_{ia}) - conv(I_{ib}) + \alpha \|_1 \tag{4}$$

where

$$label_i = \begin{cases} 1, & class(I_{ia}) = class(I_{ib}) \\ -1, & class(I_{ia}) \neq class(I_{ib}) \end{cases}$$

We use the $l_1 - loss$ for sparsity. Each ShallowID is a 1000 dimensional vector, which is capable of incorporating sufficient information for verification. What is more, the sparse structure can accelerate the whole structure works and reduce the probability of over-fitting.

### 4.4. Pre-train and fine tune are both necessary

Compared with other convolutional network for face verification, the most important part of our work is the pre-train and fine tune strategy. We use a large amount of data to train the model to learn a coarse feature space. The experiment confirms the necessity of this step.

Fig. 6 visualizes the training process adopting both pre-train and fine-tune strategy in contrast to a process with only a verification-based training step. As shown in the graph, the two curves are close in the beginning, and the model trained with only a verification step appears to be of higher accuracy than the pre-trained model. When time reaches 200 h, we operate the fine-tune. The accuracy of fine-tuned model gradually catches up with and then surpasses the other model.

A possible explanation of this phenomenon is as follows: features that can mainly represent multiple classes are learned at the stage of pre-train. Besides, thanks to the large amount of classes, the learnt model is quite discriminatory and sparse. This inter-specific competition process reinforces the robustness of model. At the stage of fine-tune, most changes occur at the last layer.

## 5. Experiments

Our experiments are conducted on two databases: LFW and AT&T. First, we will introduce the datasets we use. After that, we will test our algorithm in single sample classification. We will also test our model for verification. The efficiency test will be shown in the end.

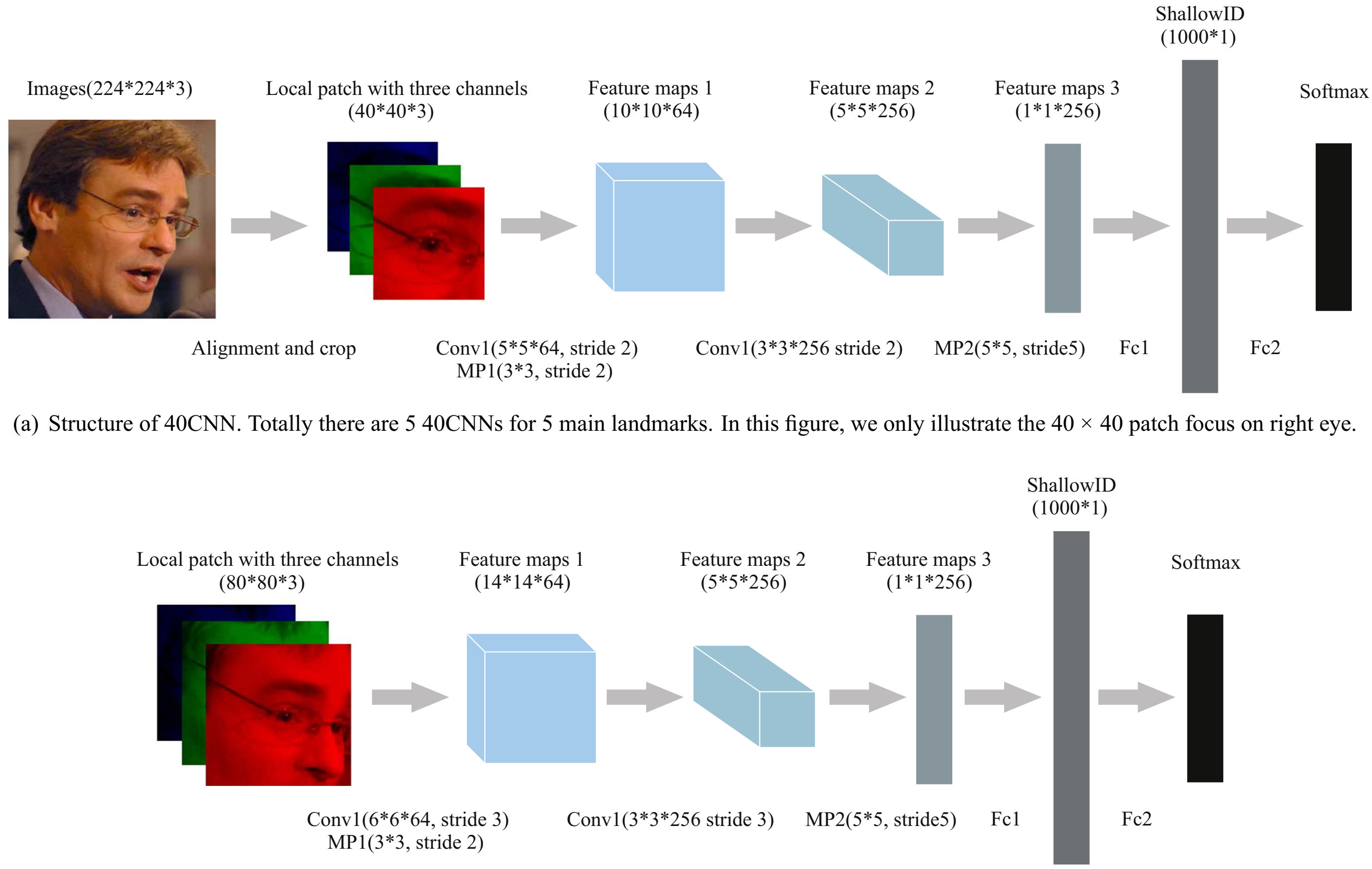
### 5.1. Face database

#### 5.1.1. Labeled faces in the wild

LFW is a facial verification standard library. Few algorithms test facial identification on that without extra training data. The reasons are as follows. First of all, LFW contains face images collected from the Internet with large variations in environment, illumination, pose and expression. Second, LFW includes 5749 subjects, only 1680 of whom have more than one image while the one with the largest amount of images has 530 images.

#### 5.1.2. AT&T database

In order to compare with other algorithms, we perform experiments on the AT&T. AT&T contains faces taken at the Olivetti Research Laboratory in Cambridge between April 1992 and April 1994. It has 36



(a) Structure of 40CNN. Totally there are 5 40CNNs for 5 main landmarks. In this figure, we only illustrate the 40 × 40 patch focus on right eye.

(b) Structure of 80CNN. The differences between 40CNN and 80CNN are the stride sizes and convolutional kernal sizes of Conv1 and Conv2.

**Fig. 4.** Details of 40CNN and 80CNN.

males and 4 females, each with 10 images, under the condition of different poses, illumination and facial expressions.

### 5.2. *Insufficient sample classification*

For insufficient sample classification, classification tools like SVM and neural networks perform no better than simply finding the nearest neighbor. To boost the classification process and save storage, we apply Linear Discriminant Analysis (LDA) on the training data.

We conduct insufficient sample classification experiment on a mixed database of the LFW and AT&T database. Each time, we take out one sample from each class to construct the training set, and leave the rest of the dataset as the test set.

Due to the lack of samples, we propose two methods to enlarge the training set.

First, we flip all the images, then rotate them with 2°, 5°, 10° and 15°. At the mean time, we stretch them both by height and width at a randomly stretch rate between 0 and 0.2.

Second, we put forward an approach named inner-variance noise augmentation. The essence of classification and verification is to minimize the variance within classes and maximize the variance between classes to make the augment more effective. Similar to the theory of LDA [45], we define the set of space basis $S$ as below:

$$\mathrm{W} = \sum_{i=1}^{c} \sum_{x \epsilon \omega_i} (x - u_i)(x - u_i)^T \tag{5}$$

where $\omega_i$ is the $i$th sample set, $c$ is the number of the class, $u_i$ is $u_i = \frac{1}{N_i} \sum_{x \epsilon \omega_i} x$, and $x$ is the feature vector of one sample.

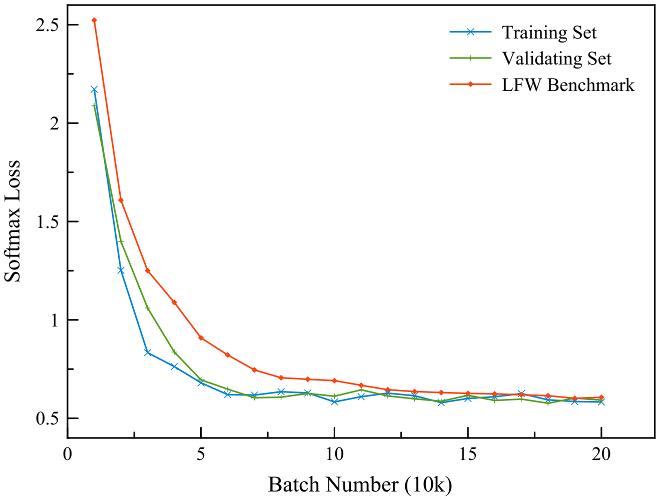


**Fig. 5.** During epoch 1–10, the training loss, valid loss and benchmark loss are all on the descent. After epoch 10, training loss and valid loss are both fluctuating within a narrow range while benchmark loss is still decenting.

$$\mathrm{B} = \sum_{i=1}^{c} N_i (u_i - u)(u_i - u)^T \tag{6}$$

where $N_i$ is the size of the $i$th class, $u$ is $u = \frac{1}{N} \sum_{x \epsilon \omega_i} N_i u_i$, $N$ is the size of all samples

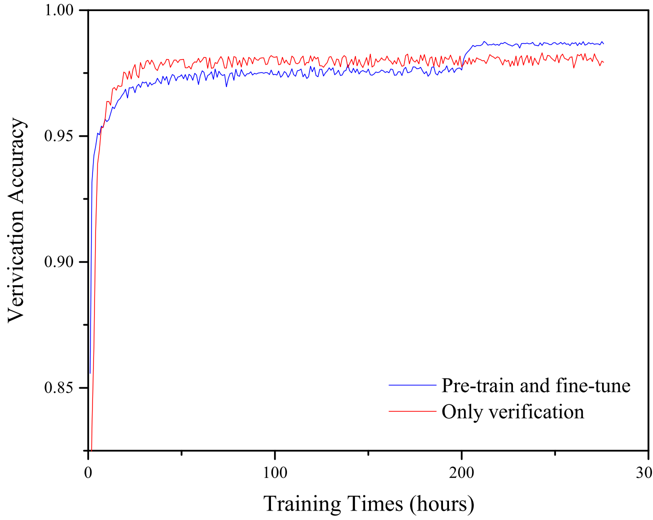


**Fig. 6.** Comparison between the pre-train using classification loss and the direct training using verification loss. We fine-tune the coarse classification of our model at the point of 200-h. The result shows that the strategy of using verification loss to fine-tune the pre-trained model achieves better accuracy on benchmark database.

$$Tx_i = \lambda_i x_i \quad (7)$$

where $x_i$ is the *i*-th feature vector of *T*. *T* is $T = W \cdot B^{-1}$

Finally, *S* is:

$$S_k = \{x_i | i \le k \text{ and } \lambda_i \text{ is the ith biggest element in } \{\lambda\}\} \quad (8)$$

the biggest *k* eigenvalues contains 80 percents of the energy of eigenvalues. Then we use the inter-class variance, drawn on the direction of these *k* eigen vectors, of each sample set needing to be extended, as the variance for sample augmentation. Then we use the inter-class variance, drawn on the direction of these *k* eigen vectors, of each sample set needing to be extended, as the variance for sample augmentation.

$$SetA^{*} = \{SetA, randnoise(\delta(A))\} \quad (9)$$

As a result, the sizes of each class become similar. Also the inter-class and intra-class variance almost remain the same, but the classification performance becomes more impartial, preventing serious over-fitting on small sample set.

We choose 80 subjects randomly in this experiment. All the subjects have no more than 100 samples, 80% of them have less than 10 samples and 50% of them only have 5 samples or less. Then we proceed classification on the original data set and the augmented set with rotation, stretch and inner-variance noise augmentation. The final size of each class is augmented to $Ex_{num}$:

$$Ex_{num} = \max(I_{max}, 100) \quad (10)$$

where $I_{max}$ is the sample number of the class with the most samples. Due to the uncertainty of inner-variance, the result of the process with augmented data is unsatisfying, sometimes even worse than the unprocessed data. But as the training set becomes larger, its advantage comes out. When the size of the maximum class exceeds 8, the difference between sizes of the classes becomes significant enough to reveal a great partiality in the classification result. However, by this sample augmentation method, the classification can still present a robust curve. We test the effectiveness of inner-var with algorithm LLB+SVM on the database. The results are shown in Table 1.

**Table 1**
Performance comparison on classification between the LLB+SVM algorithm using or not using inner-var to extend training samples.

| | No extension (%) | Extended by inner-var (%) |
|---|---|---|
| 1 | 37.908 | 35.012 |
| 2 | 42.192 | 45.763 |
| 3 | 46.997 | 48.180 |
| 4 | 53.071 | 56.801 |
| 5 | 59.937 | 58.966 |
| 6 | 64.466 | 63.887 |
| 7 | 70.210 | 72.642 |
| 8 | 73.987 | 76.971 |
| 9 | 75.001 | 79.549 |
| 10 | 77.864 | 82.385 |
| 11 | 80.012 | 85.139 |
| 12 | 81.927 | 86.074 |
| 13 | 83.296 | 88.070 |
| 15 | 81.997 | 88.821 |
| 20 | 85.076 | 90.052 |
| 100 | 82.736 | 94.932 |

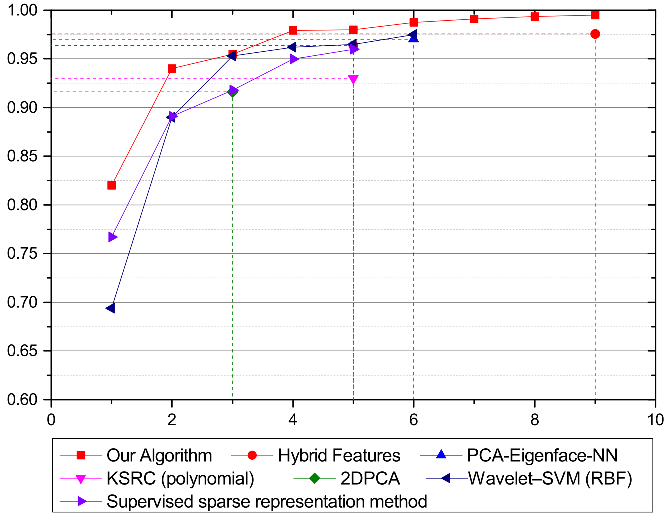


**Fig. 7.** Accuracy of different algorithms with different training sizes on AT&T face database.

### 5.3. Method comparison with insufficient samples

We compare the accuracy of our ShallowID algorithm on classification task with other algorithms with insufficient training samples to further evaluate the performance. Fig. 7 compares the average percentage identification results of other algorithms using different sizes of training sample in AT&T. Because the AT&T database has only 10 samples for each subject, the training set size for each subject ranges from 1 to 9. We randomly choose *x* samples for each subject to train model, and the other images are used to test models.

In Fig. 7, the *x*-axis denotes the training samples' size (images/subject), while the *y*-axis denotes the accuracy of identification. It could be find out that our algorithm performs better than the established state-of-the-art. The methods we compared with are from [46–51].

### 5.4. Verification

Both LLB and ShallowID methods are tested in this section. For LLB, feature vectors of LBP, LPQ and BSIF are firstly connected and compressed to 1000-dimension by SVD. After that, we apply the Joint Bayesian [25] algorithm instead of LDA for verification, in which the final compressed feature vector *x* is viewed as a

**Table 2**
Verification results on LFW.

| Method | Average Acc. |
|---|---|
| high-dim LBP [26] | 0.9517 ± 0.0113 |
| TL Joint Bayesian [52] | 0.9633 ± 0.0108 |
| DeepFace-ensemble [1] | 0.9735 ± 0.0025 |
| DeepID [18] | 0.9745 ± 0.0026 |
| ConvNet-RBM [53] | 0.9252 ± 0.0038 |
| **ShallowID** | **0.9706 ± 0.0073** |
| **Joint LLB** | **0.9479 ± 0.0132** |

summation of two independent hidden variables: $\mu$ and $\epsilon$, representing the face identity and in-class face variation respectively. An EM-like algorithm is applied to estimate $S_\mu$ and $S_\epsilon$, which are the covariance matrices of $\mu$ and $\epsilon$ respectively, and finally a log likelihood ratio $r(x_1, x_2)$ can be obtained to represent the similarity between two face images:

$$r(x_1, x_2) = \log \frac{P(x_1, x_2|H_I)}{P(x_1, x_2|H_E)} = x_i^T A x_1 + x_2^T A x_2 - 2x_1^T G x_2 \tag{11}$$

where

$$A = (S_\mu + S_\epsilon)^{-1} - (F + G) \tag{12}$$

$$\begin{pmatrix} F+G & G \\ G & F+G \end{pmatrix} = \begin{pmatrix} S_\mu + S_\epsilon & S_\mu \\ S_\mu & S_\mu + S_\epsilon \end{pmatrix}^{-1} \tag{13}$$

We apply cross-validation for training on AT&T datasets. Each round we use half of the samples in classes with more than one sample. The average accuracy of LLB method and ShallowID method is estimated to be 97.48% and 99.56%, respectively.

Meantime, we test our two methods on LFW database. For ShallowID method, an extra dataset (with about 200,000 images) is used to pre-train the network. So we follow the rule of ‘Unrestricted, label free and with outside data’ in LFW to test our model. The results are shown in Table 2.

### 5.5. Speed test

With CPU: E5-2650 v3 and GPU: Tesla K40, the joint face alignment and detection algorithm can reach the speed of 16 ms/pic (1–8 peoples @1080P). The partition, sampling and a set of three descriptors (LBP, LPQ, and BSIF) takes about 85 ms on CPU for each face while extracting ShallowID takes only 12 ms on GPU and 61 ms on CPU. For comparison, the speed of the algorithm in [26] is 80 ms per frame, tested on CPU using the configuration of 2 multi-scales and 5 facial landmarks. Our ShallowID algorithm is faster and more efficient than the LLB method.

## 6. Conclusion

In this paper, we first present a human-designed-descriptor-based feature extraction method. A joint combination of LBP, LPQ and BSIF is applied on partitions divided according to facial landmarks. This kind of feature extraction is highly invariant to illumination and gesture variance, hence fit for insufficient sample training sets.

In addition, we further propose a boosting CNN cascade framework. This algorithm runs even faster and extracts more powerful feature than the former algorithm. Its high performance is shown in LFW and AT&T databases. Also it can be applied on both classification tasks and verification tasks, needing to modify only the last layer, due to the generality and robustness of ShallowID it generates.

## Acknowledgment

This work is supported by the National Natural Science Foundation of China (Nos. 71173012 and 61472023), the project of the State Key Laboratory of Software Development Environment (No. SKLSDE-2016ZX-24) and the project in the National Science & Technology Pillar Program (No. 2014BAK06B04).

## References

[1] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, USA, 2014, pp. 1701–1708.

[2] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, arXiv preprint arXiv:1502.01852.

[3] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 2037–2041.

[4] V. Ojansivu, J. Heikkilä, Blur insensitive texture classification using local phase quantization, in: Image and Signal Processing, Cherbourg-Octeville, France, Springer, Berlin, Heidelberg, 2008, pp. 236–243.

[5] J. Kannala, E. Rahtu, Bsif: Binarized statistical image features, in: Proceedings of the International Conference on Pattern Recognition, Tsukuba, Japan, 2012, pp. 1363–1366.

[6] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 2005, pp. 886–893.

[7] G. Huang, M. Mattar, H. Lee, Learning to align from scratch, in: Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, Nevada, United States, 2012, pp. 764–772.

[8] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Active shape models-their training and application, Comput. Vis. Image Underst. 61 (1) (1995) 38–59.

[9] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[10] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of the Second IEEE Workshop on Applications of Computer Vision, Sarasota, Florida, 1994, pp. 138–142.

[11] Q. Liu, Y. Yang, R. Ji, Y. Gao, L. Yu, Cross-view down/up-sampling method for multiview depth video coding, IEEE Signal Process. Lett. 19 (5) (2012) 295–298.

[12] Y. Yang, X. Wang, T. Guan, J. Shen, L. Yu, A multi-dimensional image quality prediction model for user-generated images in social networks, Inf. Sci. 281 (October (10)) (2014) 601–610.

[13] Y. Yang, X. Wang, Q. Liu, M. Xu, L. Yu, A bundled-optimization model of multiview dense depth map synthesis for dynamic scene reconstruction, Inf. Sci. 320 (November (1)) (2015) 306–319.

[14] B. Leng, Z. Xiong, X. Fu, A 3d shape retrieval framework for 3d smart cities, Front. Comput. Sci. China 4 (3) (2010) 394–404.

[15] Y. Gao, M. Wang, D. Tao, R. Ji, Q. Dai, 3-d object retrieval and recognition with hypergraph analysis, IEEE Trans. Image Process. 21 (9) (2012) 4290–4303.

[16] B. Leng, J. Zeng, M. Yao, X. Zhang, 3d object retrieval with multi-topic model combining relevance feedback and lda model, IEEE Trans. Image Process. 24 (1) (2015) 94–105.

[17] Y. Gao, M. Wang, Z.-J. Zha, J. Shen, X. Li, X. Wu, Visual-textual joint relevance learning for tag-based social image search, IEEE Trans. Image Process. 22 (1) (2013) 363–376.

[18] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, USA, 2014, pp. 1891–1898.

[19] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Advances in neural information processing systems, Harrahs and Harveys, Lake Tahoe, 2012, pp. 1097–1105.

[20] G.E. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition, IEEE Trans. Audio Speech Lang. Process. 20 (1) (2012) 30–42.

[21] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 2008, pp. 160–167.

[22] B. Leng, X. Zhang, M. Yao, Z. Xiong, 3d object classification using deep belief networks, in: Proceedings of the 20th Anniversary International Conference on Multimedia Modeling, Dublin, Ireland, 2014, pp. II 128–139.

[23] B. Leng, X. Zhang, M. Yao, X. Zhang, A 3d model recognition mechanism based on deep boltzmann machines, Neurocomputing 151 (March (Part 2, 5)) (2015) 593–602.

[24] B. Leng, S. Guo, X. Zhang, X. Zhang, 3d object retrieval with stacked local convolutional autoencoder, Signal Process. 112 (July) (2015) 119–128.

[25] D. Chen, X. Cao, L. Wang, F. Wen, J. Sun, Bayesian face revisited: a joint formulation, in: Computer Vision-ECCV 2012, Firenze, Italy, Springer, 2012, pp. 566–579.

[26] D. Chen, X. Cao, F. Wen, J. Sun, Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, 2013, pp. 3025–3032.

[27] H.V. Nguyen, L. Bai, Cosine similarity metric learning for face verification, in: Computer Vision-ACCV 2010, Queenstown, New Zealand, Springer, 2011, pp. 709–720.

[28] Y. Sun, D. Liang, X. Wang, X. Tang, Deepid3: face recognition with very deep neural networks, arXiv preprint arXiv:1502.00873.

[29] H. Lee, R. Grosse, R. Ranganath, A.Y. Ng, Unsupervised learning of hierarchical representations with convolutional deep belief networks, Commun. ACM 54 (10) (2011) 95–103.

[30] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, arXiv preprint arXiv:1503.03832.

[31] G.E. Dahl, T.N. Sainath, G.E. Hinton, Improving deep neural networks for lvcsr using rectified linear units and dropout, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 2013, pp. 8609–8613.

[32] T.F. Cootes, G.J. Edwards, C.J. Taylor, Active appearance models, Proc. IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 681–685.

[33] X. Cao, Y. Wei, F. Wen, J. Sun, Face alignment by explicit shape regression, Int. J. Comput. Vis. 107 (2) (2014) 177–190.

[34] S. Ren, X. Cao, Y. Wei, J. Sun, Face alignment at 3000 fps via regressing local binary features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, USA, 2014, pp. 1685–1692.

[35] D. Chen, S. Ren, Y. Wei, X. Cao, J. Sun, Joint cascade face detection and alignment, in: Computer Vision-ECCV 2014, Zurich, Switzerland, Springer, 2014, pp. 109–122.

[36] G. Ghiasi, C.C. Fowlkes, Occlusion coherence: detecting and localizing occluded faces, arXiv preprint arXiv:1506.08347.

[37] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.

[38] L. Zhang, D. Samaras, Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics, IEEE Trans. Pattern Anal. Mach. Intell. 28 (3) (2006) 351–363.

[39] J. Lu, Y.-P. Tan, G. Wang, Discriminative multimanifold analysis for face recognition from a single training sample per person, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 39–51.

[40] M. Song, D. Tao, X. Huang, C. Chen, J. Bu, Three-dimensional face reconstruction from a single image by a coupled rbf network, IEEE Trans. Image Process. 21 (5) (2012) 2887–2897.

[41] E. Zhang, Y. Li, F. Zhang, A single training sample face recognition algorithm based on sample extension, in: Proceedings of the Sixth International Conference on Advanced Computational Intelligence, Hangzhou, 2013, pp. 324–327.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, arXiv preprint arXiv:1409.4842.

[43] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[44] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, USA, 2014, pp. 580–587.

[45] D. Keysers, H. Ney, Linear discriminant analysis and discriminative log-linear modeling, in: Proceedings of the 17th International Conference on International Conference on Pattern Recognition, Cambridge, UK, 2004, pp. 156–159.

[46] Y. Xu, W. Zuo, Z. Fan, Supervised sparse representation method with a heuristic strategy and face recognition experiments, Neurocomputing 79 (March (1)) (2014) 125–131.

[47] L. Zhang, W.D. Zhou, P.C. Chang, J. Liu, Z. Yan, Kernel sparse representation-based classifier, IEEE Trans. Signal Process. 60 (4) (2012) 1684–1695.

[48] E. Gumus, N. Kilic, A. Sertbas, Evaluation of face recognition techniques using pca, wavelets and svm, Expert Syst. Appl. 37 (9) (2010) 6404–6408.

[49] M. Agarwal, H. Agrawal, N. Jain, M. Kumar, Face recognition using principle component analysis, eigenface and neural network, in: Proceedings of the International Conference on Signal Acquisition and Processing, Bangalore, 2010, pp. 310–314.

[50] J. Yang, D. Zhang, A. Frangi, J. Yang, Two-dimensional pca a new approach to appearance-based face representation and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 26 (1) (2004) 131–137.

[51] K. Yesu, H. Chakravorty, P. Bhuyan, R. Hussain, Hybrid features based face recognition method using artificial neural network, in: Proceedings of the National Conference on Computational Intelligence and Signal Processing, Guwahati, Assam, 2012, pp. 40–46.

[52] X. Cao, D. Wipf, F. Wen, G. Duan, J. Sun, A practical transfer learning algorithm for face verification, in: Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 2013, pp. 3208–3215.

[53] Y. Sun, X. Wang, X. Tang, Hybrid deep learning for face verification, in: Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 2013, pp. 1489–1496.

**Biao Leng** received the B.Sc. degree from the School of Computer Science and Technology, National University of Defense Technology, Changsha, China, in 2004, and the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2009. He is an Associate Professor at the School of Computer Science and Engineering, Beihang University, Beijing, China. His current research interests include 3D model retrieval, image processing, pattern recognition, data mining and intelligent transportation systems.

**Yu Liu** is currently pursuing bachelor's degree in the School of Computer Science and Engineering in Beihang University. His research is mainly on face recognition, computer vision and deep learning.

**Kai Yu** is currently pursuing bachelor's degree in the School of Computer Science and Engineering in Beihang University. His research is mainly on face recognition and computer vision.

**Songting Xu** is currently pursuing bachelor's degree in the School of Computer Science and Engineering in Beihang University. His research is mainly on face recognition.

**Ziqing Yuan**, an undergraduate in South-central University for Nationalities, Wuhan, China, has studied in the School of economics, majoring in Finance, since September 2012. Her research interests lie in the field of data mining.

**Jingyan Qin** is the Full Professor at University of Science & Technology Beijing. She is the Ph.D. Supervisor in Big Data Information Visualization and Interaction Design at Computer Science School of USTB, and she is the Director of HCI and Design for Sustainability Research Center at Industrial Design Department, USTB. Dr. QIN is selected as the Fellow of New Century Talents Plan of Ministry of Education of China in 2013. Dr. QIN's research and education focuses on data mining, interaction design, digital entertainment design and new media art.