International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.10 (2015), pp.29-38
http://dx.doi.org/10.14257/ijsip.2015.8.10.04

Face Recognition based on Deep Neural Network

Li Xinhua,Yu Qian

Shandong Women's University
lixinhua@sdwu.edu.cn

Abstract

In modern life, we see more techniques of biometric features recognition have been used to our surrounding life, especially the applications in telephones and laptops. These biometric recognition techniques contain face recognition, fingerprint recognition and iris recognition. Our work focuses on the face recognition problem and uses a deep learning method, convolutional neural network, to solve it. And we use the Sobel operator to improve our result accuracy. LFW dataset is used for training and testing which gets a considerable result. And we also test our system on other face dataset, which also has a high accuracy on the recognition.

Keywords: *face recognition, convolutional neural network, Softmax function, Sobel operator*

1. Introduction

The definition of face recognition is a technique that, using the computer to analysis the face images, which can extract useful recognition information from them, which we always call it a feature vector, and we use it to distinguish the biology features. Recently, as the development of computer techniques, the face recognition has a larger application in our life. For example, in the public security system, it can identify the identity of the suspect; in the bank and customs control system, it can identify and prove the identity; and especially in the mobile phone application recently, due to the technology of face recognition, it helps users better on its own confidential information, and experience more secure financial transactions.

Though the face recognition has a strong ability to remember and recognize thousands of humans' faces, it is a still hard problem to computers. The main difficulties are: the diversity of face expression and the changes of age growing; images will be influenced by the illumination, the angle and distance of photographing; the difficulty of transforming 2-D image to 3-D image. Besides, face recognition mayattend to a lot fields such as image processing, computer vision, pattern recognition and neural network and it also associated tightly with brain cognition.In sum, these factors make face recognition to a challenging task.

There are thousands of papers about face recognition though it has a little developed time. And there is some usually used face dataset like MIT and CMU database. Due to the different input condition, it is quite difficult to compare those methods. So the United States Department of Defense proposed a database called FERET, which contains a common face dataset and a series of testing metric to strengthen the research and application of face recognition algorithms. This FERET dataset can be used to every method's testing and comparison contains different expression, illumination; head pose and age of different people. To our research, we use the LFW (Labeled Faces in the Wild) dataset, and this will be introduced in detail in the Section 4.

The face recognition can be summed to feature extraction from face images to get the feature vector first, and put it into a trained classifier, at last, get the class and finish the recognition task. We can summarize the methods of before to five main classes, method based on the geometrical characteristic, subspace analysis method, the method based on the elastic graph matching, hidden markov model and neural network methods. About these methods will be introduced in Section 2.

And for our work, since the four stages of face recognition task, face detection, face alignment, face representation and face classification, we use the neural network method for the representation and support the classification stage with a softmax for the last final layer. From the summation, which will be introduced in the Section 2, we get that the self-learning ability of neural network is so strong that it can get the implicit expression of information after repeating the process of learning. It has a big advantage of extracting face features though it needs a lot of input nodes and parameters, and hard for training. And at last, we show our improvement with a three channels input which two of them are computed with Sobel operators, x and y directions.

After the section 2 of related work, we describe our system structure and detail parameters setting in section 3. Section 4 is the experiments and our result. We conclude our work in section 5.

2. Related Work

There are five useful methods for face recognition summed to the past study.

First, the method based on the geometrical characteristic, which is first applied to the face recognition problem. Its basic idea is the difference of everyone's face is because of different components of every face, like the eyes, noses, mouths and jaws are different. Thus we can use the set of architectures and shapes of these components to be the features for the face recognition problem. The common algorithms are active contour model [1] and deformable template model [2].

Second, the sub-space analysis method is often used in face recognition, which contains PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis) are the two common methods. The sub-space method is to find a linear or nonlinear space transformation, which can change the original image data to a subspace, so that it make the distribution of data in the sub-space is tenser.

The most classic method is PCA-based Eigenface which was put forward by Turk [3] in 1991. This method take the face images as random variables, which turns the N× N vector of a face image to a N2 × 1 vector, and after minuses the mean data vector, uses the K-L transformation to get a set of orthogonal basis, then after keeps part of the principal components, the reduced dimension vector space of face images is got.

However, LDA [4] is aimed at the separability of the samples. It tries to find a projection direction, which can make the distance of within-class, is small and the distance of between-class is large based on the training samples' projection to that direction. Compared to the PCA method, only if the training sample is large, LDA can get a better result.

Summed on these mentioned linear sub-space methods, it is easily to find out that theyin fact propose a linearly simplification on the complicated changes of expressions, postures and illumination of the face images. So they cannot get a sufficient representation of face images. The nonlinear sub-space method uses the kernel method to implement the face recognition. Its basic idea [5] is first using a nonlinear transformation when dealing with the dataset like classifying to the linear inseparable samples, which can change the original sample space to a high dimension space (kernel space) to be linearly separable or approximate linearly separable. Thus the data sample can be classified with a linear method to carry out the nonlinear problem. This method only needs a kernel function to compute the inner product of each pair vector in the kernel space without computing the exact nonlinear transformation. The nonlinear sub-space methods have been used, such as kernel principal component analysis [6, 7], kernel Fisher Discriminant Analysis [8] and kernel independent component analysis [9] and so on.

The third method of face recognition is based on the elastic graph matching (EGM).Its basic theory is graph match, which uses a graph to represent the face, the vertices represent the local characteristics of facial points and the topological edges stand for the relationship between face features. Matching measure considers distance between the vertices and edges and it is a local characteristic matching method, which can identify the local feature points. Lades [10] proposed dynamic link structure (DLA) for face recognition: the face is composed of a group of link edge nodes which corresponding to the facial specific feature points and is known as the benchmark; the edges are represented by the distance of nodes and the nodes are represented by feature vectors contain local grey distribution information. And the similarity of face images is measured by the similarity of corresponded elastic graphs. Wiscott [11] improved this method that he used a set of image features to represent every node, enhancing the represent ability and adaptability. After these work, others continued this idea in feature analysis, reducing dimension and algorithm.

The face recognition method based on EGM takes into consideration the local details of faces and keeps the spatial information. Moreover, to a certain extent, it can ignore the deformation of changing from 3D to 2D faces. But because it essentially bundles several different frequencies of information into a single vector when extracting the face features, it is hard to extract the significant face features and the calculation costs a lot.

Fourth, researchers use the hidden markov model (HMM) to solve the problem that the different appearance of organs and the connection of each other. Based on this model, the feature observed treated as a sequence of unobserved states. Different people use different HMM parameters, and for the same person, we use the model with same parameters to represent the observed sequence of gestures and facial expressions. Samaria [12] first proposed the face model, which used a rectangular window sampling face images from top to bottom. It arranged pixels in the window into vector and used grey value as the observation vector. Nefian [13] took two-dimensional discrete cosine transform to extract the features as observation vector, reducing the storage of the parameters. And Othman [14] put forward a low complexity two-dimensional HMM, which can better describe the relationship between every organ and has a higher recognition rate. The face recognition method based on HMM allows the big change of facial expression and head rotation and gets a high recognition rate, but it costs a large computation in feature extracting and model training.

The last usually used method for face recognition is neural network (NN) to use its ability of learning and classifying for extract and recognize face features. Lin, etc. [15] use the positive and negative samples for reinforcing learning to get an ideal probability result. And they increase the learning speed by applying a modular network. This method gets a good application on face detection, face position and face recognition. Meng [16] used PCA to reduce the dimension of face samples before taking the LDA to extract discriminate features, and after these steps they proposed a RBF neural network classification. The result showed that this method had high learning efficiency and recognition performance.

Summed to all the methods from the above, all those models are hand-designed. Recently, there has been a considerable interest in deep neural network [20, 21].And a deep neural network like CNN (Convolutional Neural Network) can learn the variations from the data without any prior knowledge. The advantages of using it are: 1) it can be applied to deal with a large amount of training data, 2) from 1) it can learn a wide range of invariance exists in the data and 3) since thousands of CPU cores and GPU's [20] have been used, we can get the result with a less time. Krizhevsky et al. [20] gave us an example that large deep CNN [21] trained by standard back-propagation can achieve high recognition accuracy when trained on a large dataset.

3. System Architecture

We separate our system into three main stages, image preprocessing, face representation and face classification. So, we describe the architecture of each stage in the following three parts and we also describe our improvement in the fourth part.

3.1 Image Preprocessing

When solving a face recognition problem or other face problem, we often do the face image preprocessing first, because the original training data is not suitable for the training method. Usually, we choose to do face detection, face tracking, face cropping and face alignment. So, when comes to the face recognition problem, it is necessary to solve those problems before training.

Face detection is a computer technology that determines the locations and sizes of human faces in digital images. It detects face and ignores anything else, such as buildings, trees and bodies. Face detection can be regarded as a more general case of face localization. Researchers always believe that face detection is the first task to locate the human face among a lot of other objects in a single picture. And when faces could be located exactly in any scene, the recognition step afterwards would not be so complicated. After the face detection stage, it often comes to the face cropping progress to crop faces from the original training face images and then do the face alignment. The objective of face alignment [25] is to localize the feature points on face images such as the contour points of eye, mouth and outline. Face alignment is essential to many face processing applications including face recognition, modeling and synthesis. And after resizing, we get the input data of our model.

We use the LFW (Labeled Faces in the Wild) dataset, which will be described in detail in section 4, and there are some exiting aligned versions of it. Though the problem of aligning faces is still considered difficult, there have been shown successful result by using sophisticated techniques. These methods are in the following descriptions: (1) employing an analytical 3D model of the face, (2) searching for similar fiducial-points configurations from an external dataset to infer from, and (3) unsupervised methods that find a similarity transformation for the pixels. Based the method on the, we directly download the deep funneling images from the LFW website. And use the detection program to get the 40*48 pixels face data for our training.

3.2 Face Representation

In computer vision literature, these years, many researchers focus on the descriptor engineering. They often use the same operator to all locations in the facial image when comes to face recognition. Recently, since the training dataset becomes larger, the learning methods have started to outperform engineered features due to they can discover and optimize features for the specific task at hand [20]. Here, we use a generic representation of facial images through a large deep convolutional network.

Our convolutional neural network is trained on a muti-class face recognition task, which is to classify the identity of a face image. The net's architecture is shown in Fig 1. The input of each CNN is a preprocessed funneled gray face image of size 60 by 48 pixels and is given to a convolutional layer (C1) with 32 filters of size 5*5 (we donate this by 32*5*5@60*48). Then we get 32 feature maps and feed them to a max-pooling layer (P2) with the max over 2*2 spatial neighborhoods and the stride is 1, separately for each channel. After these is another convolutional layer (C3) that has 64 filters of size 3*3, follows with a max pooling layer (P4) which takes the max over 2*2 spatial neighborhoods with a stride of 1. The purpose of these four layers is to extract low-level features, like simple edges and texture. Though the pooling layers may cause the network to lose information about the precise position of detailed facial structure and micro-textures, we still use two layers to make the network more robust to small registration errors. We set these layers for a front-end adaptive pre-processing stage. And they are responsible for most of the computation, which expand the input into a set of simple local features.

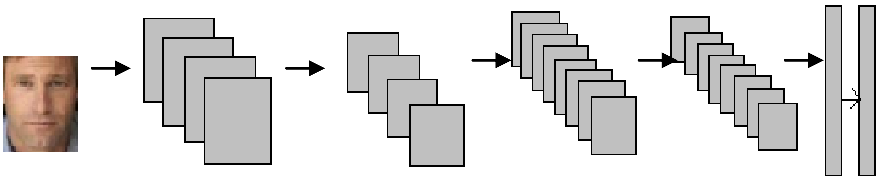

Figure 1. The Overall Architecture of our System

After the preprocessing layers, it comes two full connected layers (F5, F6) with each output unit is connected to all inputs. These layers are able to capture correlations between features captures in distant parts if the face images, such as position and shape of eyes and position and shape of mouth. After these two fully connected layers, there is a softmax layer to predict the classification result, which will be described clearly in the next part.

3.3 Face Classification

In mathematics, in particular probability theory and related fields, the softmax function, or normalized exponential, is a generalization of the logistic function that "squashes" a K-dimensional vector Z of arbitrary real values to a K-dimensional vector of real values in the range (0, 1). The function is given by

$$\sigma(\mathbf{z})_j = \frac{e^{\mathbf{z}_j}}{\sum_{k=1}^{K} e^{\mathbf{z}_k}} \quad \text{for } j=1,...,K.$$

Since the components of the vector sum to one and are all strictly between zero and one, they represent a categorical probability distribution. For this reason, the softmax function is used in various probabilistic multiclass classification methods including multinomial logistic regression: multiclass linear discriminant analysis, naive Bayes classifiers and artificial neural networks. Specifically, in multinomial logistic regression and linear discriminant analysis, the input to the function is the result of K distinct linear functions, and the predicted probability for the j'th class given a sample vector x is:

$$P(y=j|\mathbf{x}) = \frac{e^{\mathbf{x}^\mathsf{T}\mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\mathsf{T}\mathbf{w}_k}}$$

This can be seen as the composition of K linear functions and the softmax function.

$$\mathbf{x} \mapsto \mathbf{x}^\mathsf{T}\mathbf{w}_1, \ldots, \mathbf{x} \mapsto \mathbf{x}^\mathsf{T}\mathbf{w}_K$$

In neural network simulations, the softmax function is often implemented at the final layer of a network used for classification. Such networks are then trained under a log loss (or cross-entropy) regime, giving a non-linear variant of multinomial logistic regression.

Since the function maps a vector and a specific index i to a real value, the derivative needs to take the index into account:

$$\frac{\partial}{\partial q_k}\sigma(\mathbf{q}, i) = \dots = \sigma(\mathbf{q}, i)(\delta_{ik} - \sigma(\mathbf{q}, k))$$

Here, the Kronecker delta is used for simplicity (cf. the derivative of a sigmoid function, being expressed via the function itself).

The goal of training is to maximize the probability of the correct class. And we use the backward propagation method to train our network to get the low value of our loss function.

3.4 Improvement

In this part, we will introduce our improved method, which is used with a Sobel operator to compute as the two input channels. Before introduce our method’s detail, we introduce the Sobel operator first.

The Sobel operator [28], sometimes called Sobel Filter, is used in image processing and computer vision, especially within edge detection algorithms, and creates an image, which emphasizes edges and transitions. It is named after Irwin Sobel, who presented the idea of an “Isotropic 3×3 Image Gradient Operator” at a talk at the Stanford Artificial Intelligence Project (SAIP) in 1968. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. The Sobel operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation that it produces is relatively crude, in particular for high frequency variations in the image.

$$\mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A}$$

Figure 2. The Computation of Sobel Operators

The operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical. If we define A as the source image, and Gx and Gy are two images which at each point contain the horizontal and vertical derivative approximations, the computations are showed in Figure 2.

Where * here denotes the 2-dimensional convolution operation.

We use the operators of x and y directions to compute the original input image, which we will get two same size result of the origin. So we get three same size images and we put them as a three-channel input image to train our network. And the architecture is as same as we told before.

4. Experiment

Based on the architecture Section 3 gave, we develop our experiment based on the training data downloaded from the Internet. First, we do the images preprocessing, based on the LFW dataset. After getting the preprocessed face image vectors, we train our model and extract the face features. And use these reduced dimension features straight to a softmax. At last, we value our system, using the testing dataset, and also show the result with the improvement.

4.1 Face Datasets

We extract our face features from a large collection of photos from a well-known labeled dataset network, Labeled Faces in the Wild (LFW). This dataset is designed, by the compute vision lab of University of Massachusetts, Amherst for studying the problem like face recognition, face detection or face alignment. It contains more than 13,000 images of faces (right now is 13233 images) collected from the web, which have been labeled with the name of the person photographed. The whole dataset has about 5749 people in it. But there are not every person has more than one picture. There are only 1680 of them have two or more distinct photos in the dataset. The only constraint on these faces is that the Viola-Jones face detector detected them. Figure 3 is the examples of the LFW dataset.

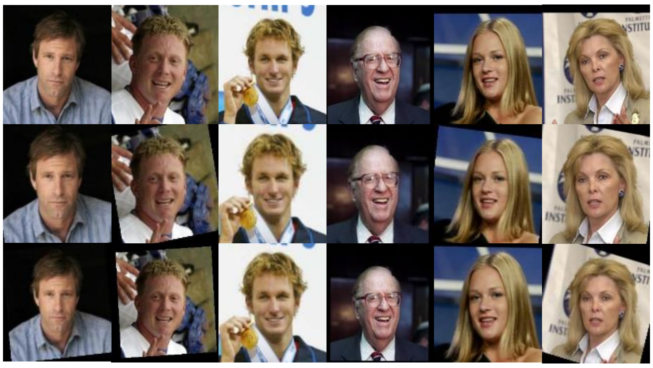

Figure 3. The Examples of LFW Dataset

4.2 Training on the LFW

We randomly select 5500 face ids for being the training set and 247 people for validation every training time. The original picture is 150*150 pixels, and we use the deep funneled images with 250*250 pixels. Since the images we download have already aligned, we then cut the images into 60*48 pixels and turn them to gray images. Before the training, we do the histogram equalization for all. However, we also compute the mean data of all the face pictures and let every picture to subtract this vector. After the image preprocessing, we get the training and testing dataset, which can be directly used to our model. We train our convolutional neural network on the 5500 picked face ids. This CNN model is trained on a GPU-based engine, implementing the standard back-propagation on feed-forward nets by stochastic gradient descent (SGD) with momentum (set to 0.9). We choose the batch size as 100 and have set an equal learning rate for all learning layers to 0.01, which was manually decreased, each time by an order of magnitude once the validation error stopped decreasing, to a final rate of 0.0001. The weights are initialized in each layer from a zero-mean Gaussian distribution with σ=0.01, and biases are set to 0.5. We train our network for 100 epochs over the whole training data which takes almost 4 days. As section 3 described, we value our system from the softmax result, which will be showed at next part.

Based on the base architecture above, we improve our method with two Sobel result's channels input, which is described clearly in the Section 3.4. We compute each gray image with x and y direction Sobel operators, noting that this gray image is processed after the histogram equalization. And compute the three new channel's mean data for image preprocessing. This architecture is as same as before, we also compare our improved result with it in the next part.

4.3. Results on the LFW

In face recognition task, recently, the computer vision community has made significant progress in unconstrained environments. The mean recognition accuracy on LFW [26] marches steadily towards the human performance of over 97.5% [27]. Given some very hard cases due to large lighting, aging effects and face pose variations in LFW, any improvement over the state-of-art is very remarkable and the system has to be composed by highly optimized modules. For our model (improved), we have achieved the accuracy among 97% and the ROC curve Figure 3 shows the results.

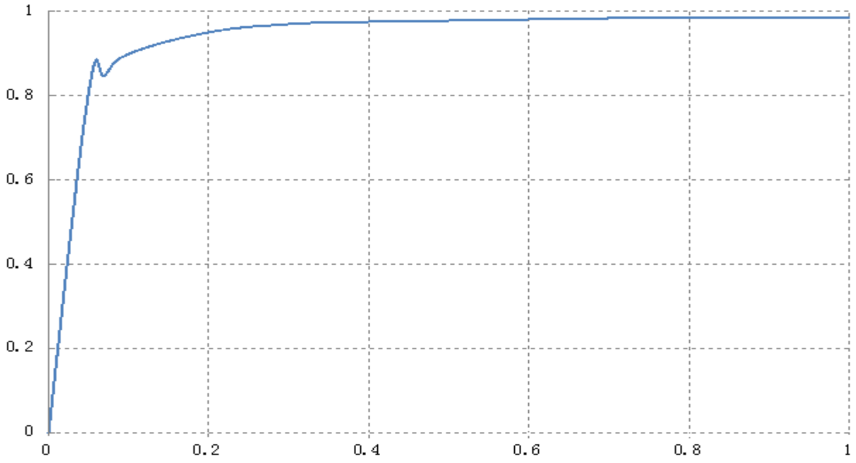

Figure 3. Model Results

5. Conclusion

We build a discriminative model for solving the face recognition task. This task always focuses on large number of classes, or all the classes are not available when training. This method we propose is a convolutional neural network to extract the face features and improve it with Sobel computation input. This work takes shorter time and a little better accuracy compared with other methods. Our future work aims at enriching our training image dataset and tries to use other loss function and find better network architectures.

References

[1] B. Olstad and A H. Torp, "Encoding of a prior information in active contour models [J]", IEEE Trans.on Pattern Analysis and Machine Intelligence, vol. 18, no. 9, (**1996**), pp. 863-872.

[2] A K. Jain and Z. Yu, "Deformable template models: A review [J]", Signal Processing, vol. 71, no. 2, (**1998**), pp. 109-129.

[3] M A Turk and A P Pentland, "Eigenfaces for recognition [J]", Journal of Cognitive Neuroscience, vol. 3, no. 1, (**1991**), pp. 71-86.

[4] J. Lu, K N. Plataniotis and A N. Venetsanopoulos, "Face recognition using LDA-based algorithms [J]", IEEETrans.onNeuralNetworks, vol. 14, no. 1, (**2003**), pp. 195-200.

[5] V N. Vapnik, "The nature of statistical learning theory", (**2004).**

[6] C. Liu, "Gabor-based kernel PCA with fractional power polynomial models for face recognition [J]", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 26, no. 5, (**2004**), pp. 572-581.

[7] J. Yang, A F Frangi and J Y. Yang, "KPCA plus LDA: A complete kernel Fisher discriminant framework for feature extraction and recognition [J]", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 27, no. 2, (**2005**), pp. 230-244.

[8] C. Liu, "Capitalizeondimensionalityincreasingtechniquesforimprovingfacerecognition grand challenge performance [J]", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, (**2006**), pp. 725-737.

[9] J. Yang, X. Gao and D. Zhang, "Kernel ICA: An alternative formulation and its application to face recognition [J]", Pattern Recognition, vol. 38, (**2005**), pp. 1784-1787.

[10] M. Lades, J C. Vorbruggen and J. Buhmann, "Distortion invariant object recognition in the dynamic link architecture [J]", IEEE Trans. on Computer, vol. 42, no. 3, (**1993**), pp. 300-311.

[11] L. Wiskott, J M. Fellous and N. Kruger, "Face recognition by elastic bunch graph matching [J]", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, (**1997**), pp. 775-779.

[12] F. Samaria, "Face recognition using hidden Markov model [D]", Cambridge, University of Cambridge, (**1994**).

[13] A. Nefian, "A hidden Markov model-based approach for face detection and recognition [D]", Georgia, Georgia Institute of Technology, (**1999**).

[14] H. Othman and T. Aboulnasr, "A separable low complexity 2D HMM with application to face recognition [J]", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 10, (**2003**), pp. 1229-1238.

[15] S H. Lin, S Y. Kung and L J. Lin, "Face recognition/detection by probabilistic decision based neural network [J]", IEEE Trans. on Neural Networks, vol. 8, no. 1, (**1997**), pp. 114-132.

[16] M J. Er, S Wu and J Lu, "Face recognition with radial basis function (RBF) neural networks [J]", IEEE Trans. On Neural Networks, vol. 13, no. 3, (**2002**), pp. 697-710.

[17] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfacesvs. fisherfaces: Recognition using class specifi c linear projection", IEEE Trans. PAMI, Special Issue on Face Recognition, vol. 19, no. 7, (**1997**) July.

[18] M. Hsuan Yang, N. Ahuja, and D. Kriegman, "Face recognition using kernel eigenfaces", In Proc. of the 2000 IEEE International Conference on Image Processing (ICIP), vol. 1, (**2000**) September, pp. 37–40.

[19] P. Y. Simard, Y. LeCun, J. S. Denker, and B. Victorri, "Transformation invariance in pattern recognition – tangent distance and tangent propagation", International Journal of Imaging Systems and Technology, vol. 11, no. 3, (**2000**).

[20] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments", In ECCV Workshop on Faces in Real-life Images, vol. 1, no. 6, (**2008**).

[21] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification", In ICCV, vol. 6, (**2009**).

[22] S. Chopra, R. Hadsell and Y. Le Cun, "Learning a similarity metric discriminatively, with application to face verification".

[23] E. Osuna, R. Freund and F. Girosi, "Training Support Vector Machines: an Application to Face Detection", (**1997).**

[24] J. Cervantes, X. Li and W. Yu, "SVM classification for large data Sets by considering models of classes distribution", (**2008).**

[25] L., Zhang, H. Ai, S. Xin, C. Huang, S. Tsukiji and S. Lao, "Robust face alignment based on local texture classifiers".

[26] G. B. Huang, M. A. Mattar, H. Lee, and E. G. Learned-Miller, "Learning to align from scratch", In NIPS, vol. 2, (**2012**), pp. 773–781.

[27] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks", In ANIPS, 2012. 1

[28] http://de.wikipedia.org/wiki/Sobel-Operator

Authors

Li Xinhua, he received the Bachelor's Degree of Science and the Master's Degree of Engineering from Shandong Normal University in 2000 and 2008 respectively. He is currently researching on Digital Image Processing and Database Theory and Application.

Yu Qian, she received the Bachelors Degree of Science from Shandong University of Technology in 2000,and the Masters Degree of Engineering from Shandong University in 2009. Her research interests include computer graphics, computer aided geometric design and machine learning.