Pose Invariant Face Recognition

Fu Jie Huang
Electrical and Computer Engineering Department Carnegie Mellon University
jhuangfu@cmu.edu

Zhihua Zhou
State Key Lab for Novel Software Technology Nanjing University
daniel@aiake1.nju.edu.cn

Hong-Jiang Zhang
Media Computing Group Microsoft Research China
hjzhang@microsoft.com

Tsuhan Chen
Electrical and Computer Engineering Department Carnegie Mellon University
tsuhan@cmu.edu

Abstract

In this paper, we describe a novel neural network architecture, which can recognize human faces with any view in a certain viewing angle range (fromy left 30 degrees to right 30 degrees out of plane rotation). View-specific eigenface analysis is used as the front-end of the system to extract features, and the neural network ensemble is used for recognition. Experimental results show that the recognition accuracy of our network ensemble is higher than conventional methods such as using a single neural network to recognize faces of a specific view.

1. Introduction

The face recognition problem has been studied for more than two decades. The approaches proposed in the literature so far can mainly be classified into two categories: model based and appearance based [1]. The model based method tries to extract geometrical parameters measuring the facial parts while the appearance based approach use the intensity or intensity-derived parameters such as eigenfaces coefficients [2][3]to recognize faces.

Due to the changes of lighting condition, expression, occlusion, rotation, etc., the human face appearance could change considerably. Recently there are many ongoing research efforts to build a face recognition system that recognizes faces rotating out of image plane. In this paper, we propose a face recognition system which can learn human faces of different views from video clips and recognize faces of any view within the rotation range.

This paper is organized as such: First, in this section, we will introduce some existing research efforts in the pose varying face recognition area, and also introduce our approach briefly. Then in section 2, we will explain our experimental setup, including the data acquisition and pre-processing. In section 3, we will introduce the front-end of our system: the view-specific eigenface analysis. In section 4, we propose a novel neural networks ensemble for recognizing human faces of difference views.

There are some existing approaches proposed to recognize faces under varying pose. One is the Active Appearance Model proposed by Cootes, etc. [4], which deforms a generic face model to fit with the input image and uses the control parameters as the feature vector to be fed to the classifier. The second approach is based on transforming an input image to the same pose as the stored prototypical faces and then useing direct template matching to recognize faces, proposed by Beymer [5], Poggio [6], and later extended by Vetter [7]. The third method is the eigenspace from all of the different views, proposed by Murase and Nayar [8], and later used by Graham and Allinson [9] in face recognition.

Our approach is also based on extending the eigenface approach. We build view-specific eigenfaces as proposed by Moghaddam and Pentland [10], i.e., to build one eigenface set for each view. Then we extract the feature coefficients of each image in the corresponding eigenspace. And we train view-specific neural networks. Each of the neural networks is trained on the feature coefficients calculated in the corresponding eigenspace. Also we build a second layer neural network to combine the decisions we get from the first layer view-specific neural networks as shown in Figure 1:

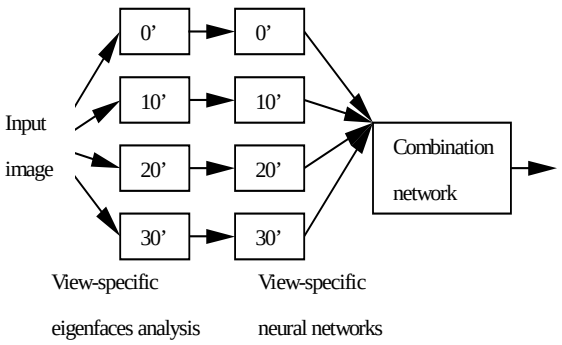

Figure 1 The System Diagram

At the test stage, when we are given one input image, we can feed it into these different channels and obtain a final decision from the second layer combination network.

Our system is designed for face recognition and key frame extraction in home video. That is to say, we want to extract all the image frames containing the specified person's face in a video sequence. The home video has the features such as large face movement and out-of-plane rotation. But at the same time, the number of the subjects to be recognized is small, usually less than ten.

2. Experimental Setup

Data acquisition In our experiment, we captured 10 video clips of 10 different subjects using a Sony DV® camcorder, and use the face region images automatically extracted from the video clips as the training data for the system. The subjects are asked to sit in front of the camcorder and rotate their head horizontally from the left side to the right side, between ± 30 degrees. We restrict the rotation range to be between +30 and –30 degrees so that in the face image both of the two eyes are always visible because the positions of the eyes will be used to align the faces in the later stage. The subjects are asked to rotate their heads continuously and smoothly between the two end points back and forth for 5 times. With the frame rate of 30 fps, we can collect different number of images depending on the speed of the rotation, with the average number as 1000 image frames.

Pre-processing First, to eliminate the effects of the non-face region variations on the recognition performance, we need to crop the face area from the whole video frame and perform recognition on the cropped face area.

Second, to calculate the view-specific eigenfaces and train the view-specific neural networks, we need to separate face images into different views. Here we group the images into 7 sets: (-35 to –25), (-25 to –15), (-15 to –5), (-5 to +5), (+5 to +15), (+15 to +25), (+25 to +35), and label images falling into each set as –30, -20, -10, 0, +10, +20, and +30 degrees, respectively. Therefore later when we say that one image is of –10 degrees, we actually mean that the pose of the face in the image is between –15 and –5 degrees.

Third, when we shoot a video clip for each subject, no special effort was made to keep the distance between the subject and the camcorder very precise. Therefore the faces of different subjects may be in different scale. We need to normalize the face images to the same scale. And for each specific view, we want to make two eyes of each subject to be located at the same positions in the normalized image.

To accomplish the requirements mentioned above, for each sequence, we use a face detector [11] to locate the face (bound by a rectangle) and the positions of the two eyes in the first image frame. Then for the following frames, we use the face tracker to track the locations of the face and eyes.

To estimate the pose of the face in the image, we use the relative location of the eyes in the face. Figure 2 shows a human head seen from above, we can calculate the distance a between the projection of the mid-point of two eyes and the center of the face, also we can obtain the radius of the head r (suppose the head has the same shape as a circle). Then we estimate the pose θ by $\theta = \arcsin(a/r)$.

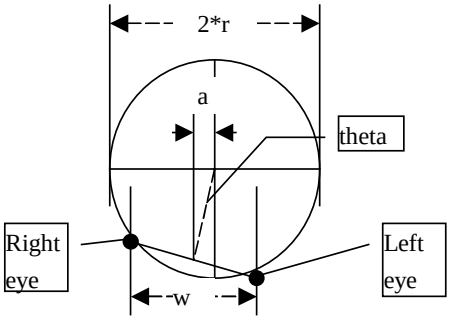

Figure 2. Pose Estimation

To normalize the face and align the eyes, we first calculate the mid-point of the two eyes, then extend from the mid-point to the left side by $w(1-\sin(\theta))/\cos(\theta)$, extend to top by $w/\cos(\theta)$, and crop a $3w/\cos(\theta)$ by $2w/\cos(\theta)$ area as the face image, then resize it to an image of 45 by 30 pixels, as shown in Figure 3.

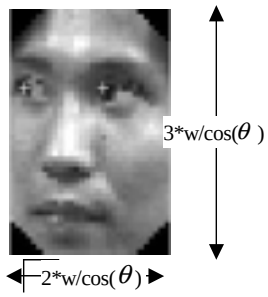

Figure 3 The Cropped Face Area

3. View-specific Eigenspace

Eigenface based approach has been used widely in the face recognition field for the frontal view face recognition, because this approach has solid mathematical background and is easy to use.

The basic idea of the eigenface approach is to represent a face image in an optimal coordinate system, or in the other words, to decompose the face image into a weighted sum of several "eigen" faces. The eigenfaces are the basis vectors of the coordinate system. The optimum criterion is that the mean-square error introduced by truncating the expansion reaches the minimum. This method is also known as Karhunen-Loeve Transformation, or Principle Component Analysis (PCA).

In our approach, to deal with the view-varying problem, we build one basis vector set for each view individually. Each basis vector set spans a specific face space. When we calculate these eigenface sets, we choose 30 images from each subject at a specific view degree, say, 0 degree, and use these 300 face images to build the eigenfaces. We compute the eigenvectors and corresponding eigenvalues of this data set, sort the eigenvalues in the descending order, and choose the first 20 eigenvectors as the eigenfaces.

As mentioned before, Murase and Nayar proposed a "universal" eigenspace by including images of different views and different objects to calculate the eigenvectors. In this universal eigenspace, the different views of one object make a "manifold", and different objects are different manifolds. Therefore it is possible to recognize both pose and identity in this space. Graham and Allinson proposed a face recognition system based on this "universal" eigenspace. But it has been shown that the that separate eigenspace will give us a better result than the global eigenspace [10].

In our system, we create individual eigenspace for each different view, that is to say, we calculate 4 eigenface sets for the 0 degree images, -10 degrees images, -20 degrees images, and -30 degrees images, respectively. One example of the eigenface set for the -20 degrees view is shown in Figure 4:

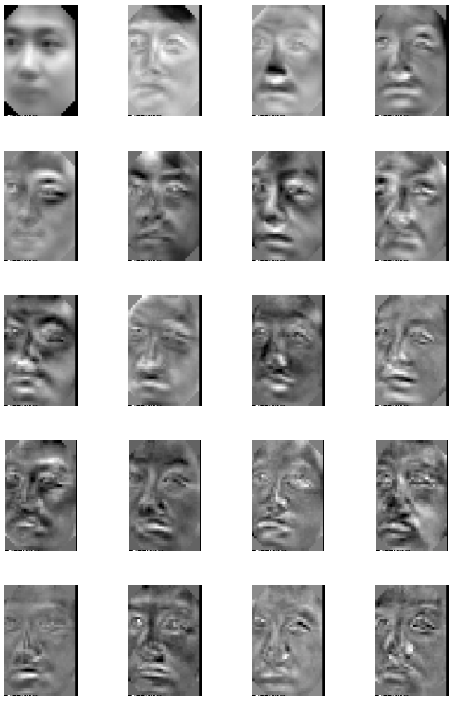

Figure 4 Eigenface Set of -20 Degrees Images

4. Recognition using Ensemble Neural Network

Neural network ensemble has been applied to many fields, such as handwritten digit recognition [12], OCR [13], speech recognition [14], seismic signals classification [15], to obtain better results than a single neural network. In face recognition field, S. Gutta and H. Wechsler [16][17] used an ensemble of RBF networks to perform face recognition.

In our system, we propose to use an ensemble neural network as the classifier to perform the pose invariant face recognition. The neural networks ensemble we proposed can be divided into two layers. The first layer contains four neural networks, each of them is a conventional feed-forward network trained with the Backpropagation (BP) algorithm based on the training data of a specific view. Each network accepts the 20 dimensional eigenface coefficients as the input vector, has 15 hidden units, and has 6 output units, as shown in Figure 5.

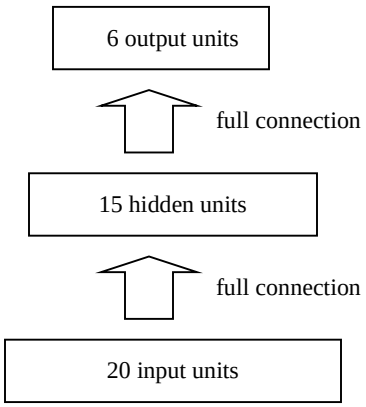

Figure 5 The First-Layer Neural Networks

The neural network has 6 output units because we only want to recognize 5 subjects in our database. We use the images of the remaining 5 subjects as a "rejection" subject, i.e., negative examples. In the test procedure, if the given image is from the first 5 subjects, the system will tell the identity of the subject, if the image is from the second 5 subjects, the system will simply reject the image, marked it as "unrecognizable".

The training data for each neural network contains 300 vectors, all of which are eigenface coefficients calculated with the corresponding view. For example, the training set of the 0 degree network is composed of 300 vectors calculated from the 0 degree face images with the 0 degree eigenfaces. Among the 300 images, each of the 5 persons to be recognized contributes 40 images, and each of the 5 persons to be rejected contributes 20 images.

The second layer is a combinational neural network trained on the output results of all the networks in the first layer.

Consider that we feed one image of a specific view into the 4 different channels, we will get 4 output vectors. Since each of the networks in the first layer has 6 output units, the input to the second layer network would be 24 dimensional.

Here we do a 24 to 24 mapping, using the second layer network with hidden units. During the training stage of this network, we force the output unit corresponding to the correct person and the correct pose to be 1, and force all the other units' value to be 0. The architecture of the second-layer neural network is shown in Figure 6:

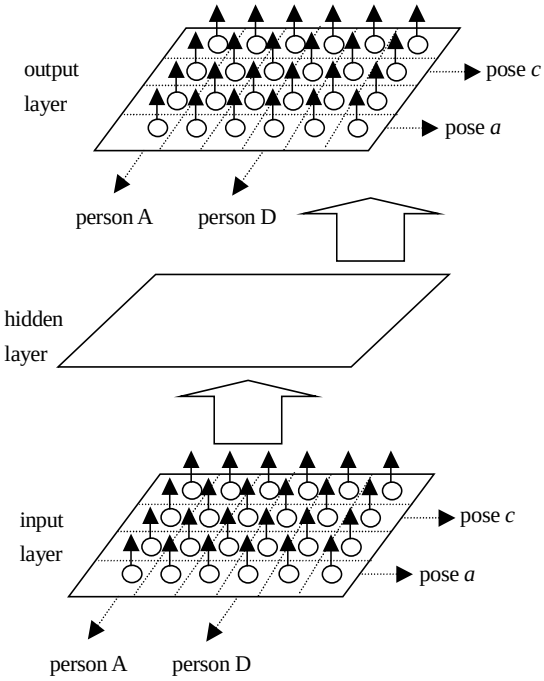

Figure 6 The Second-Layer Neural Network

Note that the input values to the second layer network are all real values, instead of binary values. That is to say, we cut off the thresholding part of the output units in the first layer neural networks.

From the output of the second layer network, we can not only tell the identity of the input image, but also tell the pose of the image, which is a bonus to the face recognition.

In our experiments, the number of the hidden units of the second layer network is also 15, which is decided empirically.

The training data set of the second-layer network is composed of 1200 24-dimensional vectors. The first 300 vectors are generated as follows. First, calculate the eigenface coefficients of the 300 images of 0 degree faces using the 0 degree eigenvector set, -10 degrees eigenvector set, -20 degrees eigenvector set and –30 degrees eigenvector set, respectively. Then feed those vectors into the corresponding networks in the first layer. Thus, there will be 4 output real-value vectors from each network. Cascade those four 6-dimensional real-value vectors into a 24-dimensional real-value vector. Thus 300 training data for the second layer network is obtained. The rest 900 instances are generated in the same way, except that the original images are of –10 degrees, -20 degrees and -30 degrees, respectively.

5. Experiments

Here we test the system using face images with unknown poses. As a comparison experiment, we also test the conventional system, which uses neural networks trained on specific views.

We will see that our system, recognizing faces with unknown poses, can outperform the conventional face recognition system recognizing faces with the view to which is system is tuned. We should note that to estimate pose is not trivial, and the result of the pose estimation will affect the face recognition result considerably in the conventional face recognition systems.

First we test the system working only with images of a specific view. Here the first layer neural networks can be seen as such a system. We train four single networks, with training data of 0 degree face images, -10 degrees images, -20 degrees images and –30 degrees images, respectively. All the networks are of the 20-15-6 architecture, i.e., they have 20 input units, 15 hidden units and 6 output units.

The test set is constructed in the same way as that of the training set. Each column in Table 1 stands for the test images of a specific view, and each row shows the performance of each network tested on images with different views.

Table 1. Experimental results of single view-specific neural networks

	0’	-10’	-20’	-30’
0’ net	98%	81%	69%	53%
-10’ net	79%	96%	84%	54%
-20’ net	52%	93%	97%	72%
-30’ net	44%	68%	73%	97%

From Table 1, we can see that if there is an accurate pose estimation process and the test image is fed to the right neural network, the recognition rate is about 97% on average, as shown by the diagonal line in the table. However, if the pose estimation is noisy, then the recognition ratio will drop very fast. For example, if we feed the face images of 0 degree pose into the –30 degrees neural network, the recognition ratio will be as low as 44%.

In our system, the pose estimation can be skipped. We can feed the input image into the system directly and get the final recognition result. The learning method has been described in Section 4. The test data set is composed of 400 face images of different views. It is generated in the same way as the training set. Table 2 shows the experimental results.

Table 2 Experimental results our pose invariant system

	0’	-10’	-20’	-30’	average
Our system	98%	98%	100%	99%	98.75%

From Table 2 we can see that we can feed face images with all the poses and get almost the same recognition ratio around 98%. By comparing the experimental results in Table 2 with those in Table 1, we can see that even without knowing the pose information, the system achieves an average recognition ratio as high as 98.75%. It is exciting that the ensemble without pose estimation (98.75%) is even better than the best single neural network with accurate pose estimation (average 97%). It means the combination of the outputs from the neural networks of different views can enhance each other to make a better recognition decision.

6. Conclusion

We have proposed a system that can recognize human face with different views. Unlike other systems, our system can recognize the identity of the person and the pose of the face at the same time. Also the recognition result is proved to be better than the systems working with a specific view. The success lies in the novel architecture of the two-layer neural network.

Also, we proposed a prototype system, which has a lot of application in the home video database creation and retrieval.

7. Acknowledgement

We would like to thank the Microsoft Research China Lab to provide us an excellent working environment and a great opportunity to work with researchers in diverse areas in multimedia processing. Also we would like to thank many colleagues with whom we had a lot of inspiring talks.

8. Reference

[1] R. Brunelli and T. Poggio, “Face Recognition: Features versus Templates,” IEEE Trans PAMI.

[2] M. Kirby and L. Sirovich, “Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces,” in IEEE Trans PAMI.

[3] M. Turk and A. Pentland, “Eigenfaces for Recognition,” *Journal of Cognitive Neuroscience*, Vol 3(1), pp 71-86, 1991.

[4] T. Cootes, G. Edwards, and C. Taylor, “Active Appearance Models,” ECCV.

[5] D. Beymer, “Face Recognitio Under Varying Pose,” AI Memo 1461.

[6] D. Beymer and T. Poggio, “Face Recognition From One Model View,” Proc. Fifth Int’l Conf. Computer Vision, 1995.

[7] T. Vetter and T. Poggio, “Linear Object Classes and Image Synthesis From a Single Example Image,” IEEE Trans PAMI.

[8] S. Nayar, H. Murase, and S. Nene, “Parametric Appearance Representation,” Early Visual Learning.

[9] D. Graham, and N. Allinson, “Face Recognition from Unfamiliar Views: Subspace Methods and Pose Dependency,” IEEE Int’l Conf. On Automatic Face and Gesture Recognition, 1998.

[10] A. Pentland, B. Moghaddam, and T. Starner, “View-Based and Modular Eigenspaces for Face Recognition,” IEEE CVPR.

[11] H. Rowley, S. Baluja, and T. Kanade, “Human face detection in visual scenes,” Carnegie Mellon University, Computer Science Tech. Rep., CMU-CS-95-158R.

[12] Hansen L K, Liisberg C, Salamon P., “Ensemble Methods for Handwritten Digit Recognition,” *Proceedings of the 1992 IEEE SP Workshop*, 1992, Vol.2, 333~342.

[13] Filippi E, Costa M, Pasero E., “Multi-layer Perceptron Ensembles for Increased Performance and Fault-Tolerance in Pattern Recognition Tasks,” *Proceedings of the 1994 IEEE International Conference on Neural Networks*, 1994, Vol.5, 2901~2906.

[14] Kirkland J., “Squad-based Expert Modules for Closing Diphthong Recognition,” *Proceedings of the Second New Zealand International Two-Stream Conference on Neural Networks and Expert Systems*, 1995, 302~305.

[15] Shimshoni Y, Intrator N., “Classification of Seismic Signals by Integrating Ensembles of Neural Networks,” *IEEE Transactions on Signal Processing*, 1998, 46(5): 1194~1201.

[16] Gutta S, Wechsler H., “Face Recognition Using Hybrid Classifier Systems,” *Proceedings of the 1996 IEEE International Conference on Neural Networks*, 1996, Vol.2, 1017~1022.

[17]S Gutta, J Huang, B Takacs, and H. Wechsler, “Face Recognition Using Ensembles of Networks,” *Proceedings of the 13th International Conference on Pattern Recognition*, 1996, Vol.4, 50~54.