Face Recognition using Eigenfaces-Fisher Linear Discriminant and Dynamic Fuzzy Neural Network

Tangquan Qi  
School of Computer and Information Science  
Southwest China University  
Chongqing, China  
e-mail: q tq2002@swu.edu.cn

Huiwen Deng* (Corresponding author)  
Institute of Logic and Intelligence  
Southwest China University  
Chongqing, China  
e-mail: huiwend@swu.edu.cn

Weiping Hu  
Institute of Logic and Intelligence  
Southwest China University  
Chongqing, China  
e-mail: huwp2000@163.com

Abstract—In order to solve the problem of face recognition in natural illumination, a new face recognition algorithm using Eigenface-Fisher Linear Discriminant (EFLD) and Dynamic Fuzzy Neural Network (DFNN) is proposed in this paper, which can solve the dimension of feature, and deal with the problem of classification easily. In this paper, we use EFLD model to extract the face feature, which will be considered as the input of the DFNN. And the DFNN is implemented as a classifier to solve the problem of classification. The proposed algorithm has been tested on ORL face database. The experiment results show that the algorithm reduces the classification error and raises the correct recognition rate. So the algorithm works well on face database with different expression, pose and illumination.

Keywords—face recognition; eigenfaces; fisher linear discriminant; feature extraction; dynamic fuzzy neural network

I. INTRODUCTION

Face recognition is very hard to solve due to its nonlinearity. In the problem of face recognition, face image data are usually high-dimensional and large-scale, recognition has to be performed in a high-dimensional space. So it is necessary to find a dimensional reduction technique to cope the problem in a lower-dimensional space. Until now, people have presented many linear/nonlinear projection methods[1,2], such as the Eigenfaces[3], PCA (Principal Component Analysis)[3], LDA (Linear Discriminant Analysis) [4,5], Fisherfaces[6], DLDA (Direct LDA) [4,6], DCV (Discriminant Common Vectors)[7], and ICA (Independent Component Analysis) [8]. In recent work, many people use a hybrid method, which combines some linear/nonlinear projection methods, for example, combining PCA and LDA[4].

Recently, a face recognition method based on PCA + RBF has been proposed[9]. This paper describes a face identification method using a probabilistic neural net. Their system is compared to the method presented at ICCST'99, based on the Karhunen-Loeve transform for feature extraction, implemented as a classifier device, that performed the identification process. In 1992, Matsuoka K proposed the face recognition system of Wavelet + RBF[10]. Back-propagation can be considered as a nonlinear regression technique, allowing a nonlinear neural network to acquire an input/output (I/O) association using a limited number of samples chosen from a population of input an output patterns. A crucial problem on back-propagation is its generation capability. A network successfully trained for given samples is not guaranteed to provide desired associations for untrained inputs as well. Lawrence S. and Giles C. L. propose a face recognition system using CNN in 1997[11]. The system combines local image sampling, a self-organizing map neural network, and a convolutional neural network. The self-organizing map provides a quantization of the image samples into a topological space where inputs that are nearby in the original space are also nearby in the output space, thereby providing dimensionality reduction and invariance to minor changes in the image sample, and the convolutional neural network provides for partial invariance to translation, rotation, scale, and deformation. The convolutional network extracts successively larger features in a hierarchical set of layers. Shang-Hung Lin and Sun-Yuan Kung propose a face recognition system based on probabilistic decision-based on neural networks(PDBNN)[12]. The PDBNN face recognition system consists of three modules: First, a face detector finds the location of a human face in an image. Then an eye localizer determines the positions of both eyes in order to generate meaningful feature vectors. Lastly, the third module is a face recognizer. Tolba A.S. and Abu-Rezq A.N. present a system for invariant face recognition (LVQ + RBF)[13]. A combined classifier uses the generalization capabilities of both learning vector quantization (LVQ) and radial basis function (RBF) neural networks to build a representative model of a face from a variety of training patterns with different poses, details and facial expressions. Phasai T. and Arnungrusmi S. proposes the integration of moment invariant and PCA for varied-pose face recognition (PCA + moment invariant)[14]. Firstly, the global feature is extracted by PCA for determining the minimum error. If error less than threshold, system will accepts the classification result from PCA. On the other hand, the system will reject and moment invariant is used to analyze the local face such as nose and eyes.

In this work, a face recognition algorithm based on the EFLD + DFNN is presented. The face recognition algorithm consists of three stages: First, the feature extraction and dimension reduction uses the Eigenface and
FLD methods. Second, DFNN is taken as classifier to classify the face feature. The last stage is the process of recognition.

This paper is structured as follows. In section 2 we briefly introduce the Eigenface and FLD, and describe the algorithm of EFLD + DFNN. In section 3 we describe the result of experiment and in section 4 we draw conclusions.

II. RECOGNITION APPROACH

A. Eigenface-Fisher Linear Discriminant Algorithm

Indeed, the Eigenfaces[16] can be considered as one of the first approaches in this sense. An $N \times N$ image $I$ is linearized in a $N$ vector, so that it represents a point in a $N$ dimensional space. However, comparisons are not performed in this space, but a low-dimensional space is found by means of a dimensionality reduction technique. N. Peter Belhumeur, P. Joao Hespanha, David J. Kriegman develop a pattern recognition algorithm which is insensitive to large variation in lighting direction and facial expression[7].

Taking a pattern classification approach, they consider each pixel in an image as a coordinate in a high-dimensional space. They take advantage of the observation that the images of a particular face, under varying illumination but fixed pose, lie in a 3D linear subspace of the high dimensional image space, if the face is a Lambertian surface without shadowing. However, since faces are not truly Lambertian surfaces and do indeed produce self-shadowing, images will deviate from this linear surface without shadowing. However, since faces are not truly Lambertian surfaces and do indeed produce self-shadowing, images will deviate from this linear subspace. Rather than explicitly modeling this deviation, they linearly project the image into a subspace in a manner which discounts those regions of the face with large deviation.

In this section, we describe the process of EFLD algorithm. EFLD algorithm consists of Eigenfaces and FLD. Main idea behind EFLD that adopt eigenfaces to represent face images into a lower-dimensional space, simultaneously let FLD find a best subspace for classification and maximize the correlation of the distance of between-class and the distance of within-class. The basic idea of EFLD is that the face images must be centered and of the same size. The last stage is the process of classification and eigenvalues for $C$ matrix. Where $v_k$ determine linear combinations of the $M$ training set of face images from the eigenfaces $\mu_i$[15]:

\[ u_i = \sum_{k=1}^{M} u_k \phi_k, \quad k = 1,2,3,..., M \]  

Where obtain an eigenface vectors $U = [u_1, u_2, u_3, ..., u_M]$.

f) In order to find a best subspace for classification, and maximize the ratio of between-class scatter and within-class scatter, so computing between-class scatter matrix $S_b$ and within-class scatter matrix $S_w$:

\[ S_b = \sum_{i=1}^{M} \left( \bar{u}_i - \bar{u} \right) \left( \bar{u}_i - \bar{u} \right)^T \]  
\[ S_w = \sum_{i=1}^{M} \sum_{j=1}^{M} \left( \bar{u}_i - \bar{u}_j \right) \left( \bar{u}_i - \bar{u}_j \right)^T \]

Where $\bar{u} = \frac{1}{M} \sum_{i=1}^{M} u_i$, $i = 1,2,3,..., M$ is the mean of eigenface images and $\bar{u}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} u_j$, $i = 1,2,3,..., M$ is the mean of $i$ th set of eigenface. $c$ is the number of the class and $n'$ represents the number of $i$ th class.

So, the rank of the between-class scatter matrix:

\[ R(S_b) = c - 1 \]  

The rank of the within-class scatter matrix:

\[ R(S_w) = n - c \]

The best subspace $W^*$ is determined by Lagrange multiplier. Meanwhile, the subspace $W^*$ consider as the input of the dynamic fuzzy neural network.

B. Dynamic Fuzzy Neural Network Description

DFNN has dynamic characteristic. The architecture of DFNN is not predefined, but dynamic variation. Namely, there is no rules before learning the DFNN. The rules are generated continuously in learning process. The architecture of the DFNN based on extended RBF neural networks to perform TSK model, which is shown in Fig. 1. DFNN consists of five layers[15][16]. Layer 1 the node of layer 1 represents an input linguistic variable.

Layer 2 each node represents a membership function that is in the form of Gaussian function:

\[ \mu_{y}(x_i) = \exp \left[ -\frac{(x_i - c_y)^2}{\sigma_y^2} \right] \]
where \( i = 1, 2, ..., r \), \( j = 1, 2, ..., u \), and \( \mu_y \) is the \( j \)th membership function of \( x_i \), \( c_j \) is the center of the \( j \)th Gaussian membership function of \( x_i \), \( u \) is the width of the \( j \)th Gaussian membership function of \( x_i \), \( r \) is the number of input variables and \( \mu \) is the number of membership functions.

Layer 3: Each node of layer 3 represents a possible IF-part for fuzzy rules. For the \( j \)th rule \( R_j \), its output is

\[
\varphi_j = \exp \left( -\frac{1}{\sigma_j^2} \sum_{i=1}^{r} (x_i - c_{ji})^2 \right)
\]

\[
= \exp \left( -\frac{\|x - C_j\|}{\sigma_j^2} \right)
\]

where \( X = (x_1, ..., x_r) \in \mathbb{R}^r \) and \( C_j = (c_{1j}, ..., c_{rj}) \in \mathbb{R}^r \) is the center of the \( j \)th RBF unit. From Eq. (11), we can see that each node in this layer also represents an RBF unit. In the sequel, RBF nodes are always used to represent rules without interpretation.

Layer 4: We call these nodes as \( N \) nodes. Obviously, the number of \( N \) nodes is equal to that of \( R \) nodes. The output of \( N_j \) is

\[
\varphi_j = \frac{\varphi_j}{\sum_{i=1}^{u} \varphi_i}, \quad j = 1, 2, ..., u
\]  

Layer 5: Each node of layer 5 represents an output variable that is the summation of input signals:

\[
y(X) = \sum_{i=1}^{u} w_k \varphi_i
\]

where \( y \) is the output variable and \( w_k \) is the THEN-part or connection weight of the \( k \)th rule.

For the TSK model:

\[
w_k = \alpha_{k0} + \alpha_{k1}x_1 + ... + \alpha_{k2}x_r
\]  

where \( k = 1, 2, ..., u \).

C. Algorithm Architecture

In this work, a face recognition algorithm based on the EFLD+DFNN is presented. The face recognition algorithm consists of three stages: First, the feature extraction and dimension reduction use the Eigenface and FLD methods. Second, DFNN is taken as classifier to classify the representations. Third, the last stage is the process of recognition. The architecture of the algorithm is shown in Fig. 2. The summary of the EFLD + DFNN algorithm is illustrated in Table I.

This algorithm has good capability of generalization, and can effectively reduce the dimensional of classification. Meanwhile, this algorithm can also reduce the computational complexity. DFNN can solve the problem of overfitting and overtraining.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Obtain the face images</td>
</tr>
<tr>
<td>2</td>
<td>Extraction and dimension reduction</td>
</tr>
<tr>
<td>3</td>
<td>Initialization DFNN</td>
</tr>
<tr>
<td>4</td>
<td>Training the DFNN</td>
</tr>
<tr>
<td>5</td>
<td>Recognition</td>
</tr>
</tbody>
</table>

**Algorithm:**

- Step 1: Obtain the face images
- Step 2: Extraction and dimension reduction
  - Where, extract feature and reduce dimension using the Eigenface and FLD algorithm.
    - 1) Computing Eigenface space, Eq. (4)
    - 2) Finding the best subspace for classification, Eq. (9)
- Step 3: Initialization of DFNN
- Step 4: Training the DFNN
  - If the set of face image is the set of training
    - Training the DFNN
  - Else
    - Jump to Step 5
- Step 5: Recognition
  - If the face images is known face
    - Output true
  - Else
    - Jump to Step 2
- Step 6: Output the result of recognition

**Table I.** Summary of EFLD+DFNN Algorithm

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Obtaining the face images</td>
</tr>
<tr>
<td>2</td>
<td>Extraction and dimension reduction</td>
</tr>
<tr>
<td>3</td>
<td>Initialization DFNN</td>
</tr>
<tr>
<td>4</td>
<td>Training the DFNN</td>
</tr>
<tr>
<td>5</td>
<td>Recognition</td>
</tr>
<tr>
<td>6</td>
<td>Output the result of recognition</td>
</tr>
</tbody>
</table>

**Figure 1.** Architecture of DFNN

**Figure 2.** Flow of the EFLD+DFNN algorithm

**Figure 3.** Architecture of the DFNN classifier
In EFLD + DFNN algorithm, DFNN is taken as classifier. In the process of training, the EFLD feature is used as the input of the DFNN. Fig. 3 illustrate the implementation architecture of the training and test of DFNN classifier.

III. EXPERIMENT

The experiment uses the ORL Database of faces. Their Database of Faces, formerly 'The ORL Database of Faces', contains a set of face images taken between April 1992 and April 1994 at the lab. The database was used in the context of a face recognition project carried out in collaboration with the Speech, Vision and Robotics Group of the Cambridge University Engineering Department. There are ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling/ not smiling) and facial details (glasses/ no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position, with tilting and rotation tolerance up to 20 degree, and tolerance of up to about 10% scaly. The files are in PGM format. The size of each image is 92×112 pixels, with 256 grey levels per pixel. Fig. 4 shows part samples in ORL face database.

In this experiment, we select 20 persons and 5 images each person in random. This 100 images are implemented as the set of training. Meanwhile, we select 100 images form 20 persons for testing.

A. Experiment Results

The experiment achieves through Matlab 7.9.0(R2009b) simulation. Table II shows some parameters in EFLD algorithm. The $\Psi$ represents the mean of all training face images. $U$ represents the projected centered vectors onto eigenspace, in other word, the eigenface space. And the parameter of $W$ represents the best subspace. $V_{FLD}$ represents the largest $(c-1)$ eigenvectors of matrix $W$. Fig.5 (a) shows the feature distribution of the mean of all training face images. Fig.5 (b) displays the isolines of matrix $W$ which shows the best subspace distribution.

B. The Results of DFNN Classifier

In the training, the some parameter of DFNN must be predefined. The input of DFNN is the best subspace $w^*$, which computing in the extracting feature using Eigenfaces+FLD algorithm. Meanwhile $w^*$ is taken as the target output of DFNN to supervise the learning of DFNN. And initial the parameters of DFNN. The DFNN classifier can much better classify the training face images, so that reduce the false recognition rate in the best. The number of the unit of RBF is 46, and the number of the classification is 19.

The performance of the DFNN obtains form fig.6 (a) to (d), which show the actual error of DFNN, the fuzzy rule generation., the root square mean error and the desired actual output and input data, respectively.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value Min</th>
<th>Max</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Psi$</td>
<td>10304*1</td>
<td>50.1800</td>
<td>184.1000</td>
</tr>
<tr>
<td>$U$</td>
<td>80*100</td>
<td>-4.2754e+06</td>
<td>3.683e+06</td>
</tr>
<tr>
<td>$W$</td>
<td>19*100</td>
<td>-1.7105</td>
<td>1.9336</td>
</tr>
<tr>
<td>$V_{FLD}$</td>
<td>80*19</td>
<td>-4.0423e+07</td>
<td>6.3441e+07</td>
</tr>
</tbody>
</table>

Figure 5. (a) the mean of all training face images, (b) the output matrix of $W$.

Figure 6. (a) the actual output error, (b) the fuzzy rule generation, (c)the root mean squared error (RMSE), (d) the desired actual output and the input data.
TABLE III. the result of comparison of the performance, the ‘-’ represents unknown the number of simulation.

<table>
<thead>
<tr>
<th>Methods</th>
<th>The number of simulation</th>
<th>$E_{\text{ave}}$ / %</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA + Fisher + RBF[17]</td>
<td>6</td>
<td>1.92</td>
</tr>
<tr>
<td>M-PCA[18]</td>
<td>10</td>
<td>2.4</td>
</tr>
<tr>
<td>PCA + RBF[9]</td>
<td>-</td>
<td>4.9</td>
</tr>
<tr>
<td>Wavelet + RBF[10]</td>
<td>-</td>
<td>3.7</td>
</tr>
<tr>
<td>EFLD + DFNN</td>
<td>3</td>
<td>1.80</td>
</tr>
</tbody>
</table>

D. Comparison of the Performance

EFLD+DFNN algorithm is detected using ORL face database. In experiment, to judge the performance of algorithm uses the average false error (AFE). The so-called AFE $E_{\text{ave}}$ is defined by the below formulation:

$$E_{\text{ave}} = \frac{\sum_{i=1}^{q} \frac{n_{\text{mis}}}{q_{\text{tot}}}}{q}$$

(15)

Where, $q$ represents the number of experiments, and $n_{\text{mis}}$ is the number of wrong classification in the $i$th epoch. And $q_{\text{tot}}$ represents the total number of testing sample. Based on $E_{\text{ave}}$ rule, the result of comparison of the performance in the same ORL face database is shown in Table III. In this paper, the EFLD + DFNN algorithm $E_{\text{ave}}$ is 2.42.

IV. CONCLUSION

The EFLD + DFNN algorithm was proposed in this paper. The algorithm consists of three steps. In the first step, the dimension reduction use Eigenface algorithm and finds the best subspace for the classification. In the second step, DFNN is implemented as a classifier. The last step is recognition. The experiment result shown that the EFLD+DFNN algorithm works well on face database with different expression, pose and illumination.

To deal with the problem caused by natural illumination variation, some modification of the algorithm is required. And it will be the future work.

ACKNOWLEDGMENT

This paper is supported by the Key Project of Chongqing(project no. 08jwsk277).

REFERENCES


