Face recognition based on artificial immune networks and principal component analysis with single training image per person

Guan-Chun Luh, Department of Mechanical Engineering, Tatung University, Taiwan, ROC, gluh@ttu.edu.tw

Abstract

Various methods could deal well with frontal view face recognition if there were sufficient number of representative training samples. However, few of them worked well if only single training image per person was available. This study proposes a face recognition method based on artificial immune networks and principal component analysis to solve the one training sample problem. The performance of the present method was evaluated utilizing the ORL face database. The results show that this method gains higher recognition rate in contrast with most of the developed methods.

Keywords: Face recognition; artificial immune networks; one training sample problem; principal component analysis; ORL

1. Introduction

Face recognition has received significant interest and attention from numerous researchers in the pattern recognition field over the last two decades. Face recognition is nowadays an important issue due to its wide range of potential applications related to biometrics, information security, identity authentication, law enforcement, video surveillance, smart cards, access control systems and so forth [1-3]. The purpose of face recognition is to recognize or authenticate one or more persons from still/video images of a scene from a stored database of faces. Face recognition has several advantages over the other biometric technologies since it is natural, non-intrusive, and easy to use. People can access it in a comfortable way without direct contact. Additionally, it can be employed ubiquitously in numerous applications. Recent surveys of face recognition techniques can be found in literature [3, 4].

A face recognition system usually contains four modules: detection, alignment, feature extraction, and matching. Its performance relies highly on the extracted features to characterize the face pattern and the classification methods to make a differentiation between faces. Essentially, the face representation was attained by means of two categories, appearance-based and feature-based approaches. The former applies holistic texture features and is used to the face or specific region of it while the latter employs the geometric relationship between the facial features such as eyes, nose, and mouth. Amongst the appearance-based approach, principal components analysis (PCA) [5] is a famous dimension reduction and feature extraction method that has been effectively and extensively utilized in pattern recognition, computer vision and signal processing. PCA algorithm finds an optimal linear transformation based on Karhunen–Loève transform to reduce the dimensionality of a vector by approximating it with most major eigenvectors. Sirovich and Kirby [6] first applied PCA for efficient representation of a face image. Subsequently Turk and Pentland [7] developed the well-known Eigenface for face recognition in 1991. Since then, PCA relative face recognition schemes including 2DPCA, 2D(PC)A, (2D)2PCA, DCT-PCA, and 2DKPCA have been comprehensively investigated. PCA based face recognition typically contains two stages: training and classification. In the training phase, an eigenspace is established from the training samples and then mapped to the eigenspace for classification. As to the classification phase, an input face is projected to the same eigenspace and subsequently classified by appropriate classifiers.

Face recognition is a difficult problem due to several image variations in real-life including illumination, lighting, facial expression, partial occlusion and imprecise localization of face area. The typical approach to handle these variations is employing large and representative training sample sets. However, there is typically barely one training sample per person in many applications such as law enforcement, identity card or passport verification. This kind of
realistic “one sample per person problem” severely challenges existing face recognition techniques, especially their robustness performances under possible variations. It has rapidly emerged as an active research sub-area in recent years. Although several methods [8-16] have been proposed dealing with the one sample problem, the variation issue is far from solved. Recent surveys of face recognition techniques employing one training image can be found in literatures [17].

In this study, a face recognition architecture based on PCA followed by artificial immune networks called PCA-IN was proposed to solve the single training sample problem. The recognition mechanism consists of two phases: the PCA feature extraction phase and the immune network classification phase utilized as a collection of individual classifiers. Figure 1 shows the schematic diagram for the proposed method. In PCA-IN, each person has his/her corresponding immune classifier containing a number of antibodies. PCA-IN is trained (optimized) employing genetic algorithms (GAs) before face recognition implementation. GAs is based on the mechanism of natural selection and evolution and has been utilized in searching for the global optimum for many applications. It combines survival of the fittest individual among population with a structured and randomized information exchange to form a search algorithm with some of the innovative flair of human search. To evaluate the performance of the PCA-IN, experimental study is carried out on the ORL laboratories database.

The rest of the paper is organized as follows: PCA is introduced in Section 2; artificial immune network is described in Section 3; Section 4 presents the PCA-immune network technique. Finally, Sections 5 and 6 present the experimental results, discussions and conclusions.

2. Feature extraction using principal component analysis

Principal component analysis is a usually utilized dimensionality reduction technique employing in face recognition to transform several possibly correlated variables into a smaller number of uncorrelated principal components. The main process for feature extraction employing PCA can be stated as follows:

A two-dimensional face image with $N \times N$ size can be considered as a vector of dimension $N^2$. The main idea of the principle component is to find the vectors that best account for the distribution of face images within the entire image space. These vectors define the subspace of face images called “face space”. Each of these vectors is a linear combination of the original face images. They are “eigenfaces” since these vectors are the eigenvectors of the covariance matrix corresponding to the original face images.

Let the training set of face images be $\Gamma_1, \Gamma_2, ..., \Gamma_M$, then the average of the set is defined as follows,

$$\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i$$
Consequently, each face differs from the average by the vector $\Phi_i - \Psi_i^{\ell}$. This set of very large vectors is then subject to PCA to find a set of orthonormal vectors best describing the distribution of the data. The $k$th orthonormal vector, $u_k$, is selected such that

$$\lambda_k = \frac{1}{M} \sum_{i=1}^{M} (u_i^T \Phi_i)^2$$

is a maximum, subject to

$$u_i^T u_i = \begin{cases} 1, & \text{if } \ell = k \\ 0, & \text{otherwise} \end{cases}$$

The vectors $u_i$ and the corresponding scalars $\lambda_i$ are the respective eigenvectors and eigenvalues of the covariance matrix

$$C = \frac{1}{M} \sum_{i=1}^{M} [\Phi_i \Phi_i^T] = AA^T$$

where the matrix $A = [\Phi_1, \Phi_2, \ldots, \Phi_M]$. Since the covariance matrix $C$ is an $N^2 \times N^2$ real symmetric matrix, it will be an intractable job to calculate the whole $N^2$ eigenvectors and eigenvalues for typical image sizes. Accordingly a computationally feasible method to find these eigenvectors is necessary. Consider the eigenvectors $v_i$ of $A^T A$ such that $A^T A v_i = \mu_i v_i$. Premultiplying both sides by matrix $A$, we have

$$A^T A A v_i = \mu_i A v_i$$

Obviously, $A v_i$ are the eigenvectors and $\mu_i$ are the corresponding eigenvalues of covariance matrix $C$. We can then construct the $M \times M$ matrix $L = A^T A$, and derive the $M$ eigenvectors of $L$. These vectors determine linear combinations of the $M$ training set face images to form the eigenfaces $u_\ell M_k 1, 2, \ldots, M$$

The calculations will be thus greatly reduced from the order of the number of pixels in the images ($N^2$) to the order of the number of images in the training set ($M$). The calculations become quite convenient since the training set of face images is relatively small ($M < N^2$) in realistic applications. Moreover, the associated eigenvalues allow us to rank the eigenvectors according to their usefulness in characterizing the variation among the images. The eigenface images calculated from the eigenvectors of $L$ span a basis set that can be used to describe face images.

In real world application, smaller number of eigenfaces is adequate for face identification since accurate reconstruction of the image is not necessary. In the task of face recognition, the eigenfaces span an $n$ ($n < M$) dimensional subspace of the original $N^2$ image space are sufficient for reliable representation of the faces and the $n$ significant eigenvectors of the $L$ matrix are chosen as those with the largest associated eigenvalues. After that, any new input face image ($\Gamma_{n\text{ew}}$) can be transformed into its eigenface components as follows,

$$v_i = \sum_{k=1}^{n} \omega_k \Phi_i$$

The weights form a projection vector $\Omega = [\omega_1, \omega_2, \ldots, \omega_n]$ which describes the contribution of each eigenface in representing the input face image and treats the eigenfaces as a basis set for face images. Consequently, classification can be achieved by comparing the projection vectors of the training face images with that of the input face image based on the Euclidean Distance between the face classes and the input face image shown below,

$$e_i = ||\Omega - \Omega_i||$$

3. Artificial immune system

The natural immune system protects living bodies from the invading of foreign substances, called antigens, including viruses, bacteria, and other parasites. Lymphocytes float freely in blood and lymph nodes, and patrol everywhere for antigens, then gradually drift back into the lymphatic system, to begin the cycle all over again. There are mainly two types of lymphocytes, namely B-cells and T-cells, which play an important role in immunities. The former takes part
in the humoral immunity that secretes antibodies (Abs) by clonal proliferation, and the latter
takes part in cell mediated immunity. One class of T-cells, called Killer T-cell, destroys the
infected cell whenever they recognize the infection. The other class which triggers clonal
expansion and stimulate/suppress antibody formation is called Helper T-cell. Figure 2 shows the
model describing the relationship between the major components in the immune system.

![Figure 2. Illustration of the biological immune system](image)

Once an infectious foreign pathogen assaults the human body, the macrophage has surface
receptors to detect and destroy the invader. Then the macrophage turns into an Antigen
Presenting Cell (APC). The APC interprets the antigen appendage and extracts the associated
features, by processing and presenting antigenic peptides on its surface to T-cell and B-cell.
These antigenic peptides are kinds of molecules called MHC (Major Histocompatibility
Complex) used to differentiate a “self” from other “non-self” (antigen). These lymphocytes will
be able to sensitize this antigen and be triggered. Afterward the Helper T-cell releases the
interleukines, the stimulation or suppression signals, and acting on the cells. In the other hand,
B-cell becomes stimulated when an antibody binds to an antigen. In addition, B-cells are as well
affected by the Helper T-cells during the immune responses. The Helper T-cell plays a
extraordinary key role for deciding the immune system to the cell mediated immunity or the
humoral immunity, and connects the non-specific immune response to make a more efficiency
specific immune response.

Affinity maturation occurs when the maturation rate of a B-cell clone increases in response
to a match between the clone’s antibody and an antigen. Subsequently, those mutant cells are
bound more tightly and stimulated to divide more rapidly. It has been demonstrated that the
immune system has the capability to recognize foreign pathogens, learn and memorize, process
information, and discriminate between self and non-self [18]. In addition, the immunity can be
maintained even faced with a dynamically changing environment. The biological immune
system is generally viewed as a mechanism of a highly adaptive, learning, distributed, and
detection system. In addition, it can recognize different pathogen patterns and generate selective
immune responses. Recognition is achieved by inter-cellular binding, which is determined by
molecule shape and electrostatic charge. Hence, B-cell becomes stimulated when an antibody
receptor binds to an antigen. Antibodies have the capability of binding pathogens that they have
never learned to recognize. This kind of anticipatory capability is due to a broad coverage of
pathogen space realized by the antibody receptors produced by the immune system.

The concepts of the artificial immune system (AIS) are inspired by ideas, processes, and
components, which extracted from the biological immune system. There are abundance source
of theories in immune system, which can play an important role in giving insights for
engineering or computer based methods. It has been demonstrated that the learning and adaptive
capabilities of AIS have a great potential in the fields of machine learning, computer science
and engineering. Amongst Jerne [19] has proposed an idiotypic network hypothesis based on
mutual stimulus and suppression between antibodies as Figure 3 shown. This hypothesis is
modeled as a differential equation simulating the concentration of a set of lymphocytes. The
concept of immune network states that the network dynamically maintains the memory using a
feedback mechanism within the network. Jerne concluded that the immune system is similar to
the nervous system when viewed as a functional network. Based on Jerne’s immune network hypothesis, Farmer et al. [20] proposed the following equations calculating the variation on the concentration of antibodies in immune networks.

\[
\frac{dA_i(t)}{dt} = \left( \sum_{j=1}^{N_A} m_{ij}^u a_j(t) - \sum_{k=1}^{N_A} m_{ik}^s a_i(t) + m_i - k_i \right) a_i(t)
\]

\[
a_i(t) = \frac{1}{1 + \exp(0.5 - A_i(t))}
\]

where \(i, j = 0, 1, \ldots, N_A\) are the subscripts to distinguish the antibody types and \(N_A\) is the number of antibodies. \(A_i\) and \(a_i\) are the stimulus and concentration of the \(i\)th antibody. \(m_{ij}^u\), \(m_{ik}^s\) indicate the stimulative and suppressive affinity between the \(i\)th and the \(j\)th, \(k\)th antibodies, respectively. \(m_i\) denotes the affinity among antigen and the \(i\)th antibody, and \(k_i\) represents the natural death coefficient.

On the other hand, Hightower et al. [21] suggested that all possible antigens could be declared as a group of set points in an “antigen space” and antigen molecules with similar shapes occupy neighboring points in that space. It indicates that an antibody molecule can recognize some set of antigens and consequently covers some portion of antigen space as Figure 4 illustrated. The collective immune response of the immune network is represented as

\[
\sum_{i=1}^{N_A} f(Ab_i)
\]

where \(f(Ab_i)\) indicates the immune response function between antigen and the \(i\)th antibody. Note that any antigen in the overlapping converge could be recognized by several different antibodies simultaneously.
4. PCA based artificial immune networks

In the proposed PCA-IN scheme, the antigen space is defined as the data set of \( n \)-dimensional eigenvectors \( \{ v_i \}_{i=1}^{N_{Ag}} \), \( v_i \in \mathbb{R}^n \); \( N_{Ag} \) indicates the number of antigens/training face images. For the purpose of efficiently neutralizing the antigen, different antigens require qualitatively different immune responses. The antigenic context has become correlated with the associated appropriate types of immune response. Here the antibody is defined as \( \{ p_j \}_{j=1}^{N_{Ab}} \), \( p_j \in \mathbb{R}^n \); \( N_{Ab} \) indicates the number of antibodies in each immune network classifier. The pseudo-code for the PCA-IN algorithm is listed in Table 1 and described in the following steps.

### Table 1 Pseudo-code of the proposed PCA-IN

<table>
<thead>
<tr>
<th>procedure</th>
<th>PCA-IN_face recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{Ag} )</td>
<td>Number of antigens/training face images;</td>
</tr>
<tr>
<td>( N_{Ab} )</td>
<td>Number of antibodies in each immune network classifiers;</td>
</tr>
<tr>
<td>( n )</td>
<td>Number of eigenfaces;</td>
</tr>
<tr>
<td>( N_{pop} )</td>
<td>Number of population size;</td>
</tr>
<tr>
<td>( N_{iter} )</td>
<td>Stopped iteration of PCA-IN;</td>
</tr>
<tr>
<td>( P_c )</td>
<td>Crossover rate;</td>
</tr>
<tr>
<td>( P_m )</td>
<td>Mutation rate;</td>
</tr>
</tbody>
</table>

### procedure training/optimizing PCA-IN employing genetic algorithms

Set \( k = 0 \);
Randomly selected training face images;
Randomly initialize immune network classifiers;
Randomly initialize distribution coefficient;
Generate the initial individual/chromosome;
Calculate the affinity value;
Calculate the fitness value;

while \(( k < N\_iter )\)
Selection;
Crossover;
Mutation;
Calculate the fitness value of offspring population;
Update and save the best individual;
end while
end procedure

**Step 1 Random selection of the train face images/antigens**

Training face image for the \( i \)th person is randomly selected and then converted to an \( N_{Ag} \times n \) dimensional eigenface

\[
v_i = \begin{bmatrix} v_{i,11} & v_{i,21} & \cdots & v_{i,N_{Ag},1} \\ v_{i,12} & v_{i,22} & \cdots & v_{i,N_{Ag},2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{i,1n} & v_{i,2n} & \cdots & v_{i,N_{Ag},n} \end{bmatrix}
\]  

(2)

where \( N_{Ag} \) is the number of training face images and \( n \) is the number of eigenvector.

**Step 2 Random initialization of immune network classifiers**

Antibodies of an immune network classifier for the \( j \)th person are initialized randomly. Immune classifier for each person is characterized by a \( N_{Ab} \times n \) matrix,
where \( N_{Ab} \) is the number of antibodies in each immune classifier.

**Step 3 Chromosome representation of genetic algorithms**

Genetic algorithm is utilized to optimize the PCA-IN and the corresponding individual/chromosome for the \( j \)th immune network classifier is defined as the following \( N_{Ab} \times (n+1) \) matrix,

\[
IN_j = \begin{bmatrix}
\beta_{j,1} & \beta_{j,2} & \ldots & \beta_{j,N_{Ab}} \\
\beta_{j,1} & \beta_{j,2} & \ldots & \beta_{j,N_{Ab}} \\
\vdots & \vdots & \ddots & \vdots \\
\beta_{j,1} & \beta_{j,2} & \ldots & \beta_{j,N_{Ab}} \\
\end{bmatrix}
\]

where \( \beta \) is the distribution factors of antibodies in the \( j \)th immune classifier.

**Step 4 Calculate the affinity value between antigens and antibodies**

The affinity \( m_{ij} \), defined as the matching ratio between the \( k \)th antigen epitope and the \( \ell \)th antibody of each immune classifier, is characterized as a Boltzmann-Gibbs distribution function [14] in this paper.

\[
m_{ij} = \frac{e^{-d_{ij}^2}}{Z}
\]

Where \( \beta \) is a parameter controlling the distribution shape, and \( d_{ij} = ||v_k - p_{\ell}|| \) is the Euclidean distance between the \( k \)th antigen epitope vector and \( \ell \)th antibody vector. The normalizing factor \( Z = \sum_{k=1}^{N_{Ab}} e^{-d_{ij}^2} \) is called the partition function. Clearly the area of coverage provided by an antibody is determined by the distribution factor \( \beta \). The larger the value of \( \beta \), the smaller the coverage area occupied. In other word, the immune response becomes more specific with larger \( \beta \) value. Note that the distribution factor \( \beta \) is initialized randomly. The distance \( d_{ij} \) represents the structural similarity between antigen epitope and antibody. Smaller distance \( d_{ij} \) anticipated that the antigen epitope is more matched to the antibody’s paratope. It should be noted that each receptor gives a significant affinity only in a neighborhood near the center. The affinity decreases monotonically with distance from the center. This phenomenon allows the immune system to recognize similarity between antigens in terms of overlapping sets of epitopes, and hence use previous memory clones for induction of the appropriate types of immune response against correlated antigen.

**Step 5 Calculate the fitness value of the immune classifiers**

The response of the overall immune networks is derived by determining the set of affinities associated with the receptors and the structural similarity between antigen and antibody defined by quantification of the distance in antigen space. The collective immune response function for the \( i \)th antigen/training image of the immune networks is

\[
f(v_i) = \sum_{j=1}^{N_{Ab}} m_{ij}
\]

For the PCA-IN, the \( i \)th immune classifier tries to maximize the immune response of the self antigens (training images of the \( i \)th person) while minimize the responses to non-self antigens (the remaining training images of the other persons). The fitness function of the \( i \)th immune classifier is defined as following

\[
fit_i = w_1 \sum_{j=1}^{N_{Ab}} m_{ij} + w_2 \frac{1}{\sum_{k=1}^{N_{Ab}-N_{self}} m_{ik}}
\]
where \( N_{\text{self}} \) and \( N_{\text{non-self}} \) represent the number of self/training images of the \( i \)th person and that of the non-self/images of the remaining person, respectively. Parameters \( w_1 \) and \( w_2 \) are the associated weighting values, \( w_1 + w_2 = 1 \). Each \( i \)th immune network tries to maximize its immune response with the associate \( i \)th personal training images and minimize its immune response with the remaining training images.

**Step 6 Optimizing PCA-IN employing genetic algorithms**

The genetic operators utilized are roulette wheel selection, arithmetic crossover and Gaussian mutation with crossover and mutation rate of 0.8 and 0.05, respectively. In roulette wheel selection, individuals are given a probability of being selected that is directly proportionate to their fitness. Two individuals are then chosen randomly based on these probabilities and produce offspring. The fittest individuals have a greater chance of survival than weaker ones. This replicates nature in that fitter individuals will tend to have a better probability of survival and will go forward to form the matting pool for the next generation. However, weaker individuals are not without a chance. In nature such individuals may have genetic coding that may prove useful to future generations.

Arithmetical crossover is defined as a linear combination of two parent individuals/immune network classifiers. If \( \text{IN}_i \) and \( \text{IN}_j \) are two individuals to be crossed, the resulting offspring are

\[
\begin{align*}
\text{IN}_{i,\text{offspring}} &= w \cdot \text{IN}_i + (1-w) \cdot \text{IN}_j \\
\text{IN}_{j,\text{offspring}} &= w \cdot \text{IN}_j + (1-w) \cdot \text{IN}_i
\end{align*}
\]

(8)

(9)

where \( w \) is a random number uniformly distributed in \([0.0, 1.0]\).

Gaussian mutation consists in adding a random value from a Gaussian distribution to each element of an individual's matrix to create a new offspring as following

\[
\text{IN}_{\text{offspring}} = (1 + N(-1,1)) \cdot \text{IN}
\]

(10)

where \( N(-1,1) \) is a random number distributed in \([-1, 1]\).

**Step 7 Stopping criteria**

The training of PCA-IN stops optimizing once a predefined number of iterations, \( N_{\text{iter}} \), is reached. Otherwise, the procedure will be repeated from steps 4 to step 6.

**5. Experiment results and discussions**

To evaluate the performance of the proposed PCA-IN, experimental studies are carried out on the AT&T laboratories database of faces (formerly also referred to as “The ORL database”) images of Cambridge University. The ORL database contains 400 face images from 40 individuals (4 female and 36 male) captured over the span of a 2-year period from subjects aged between 18 and 81. The total number of images for each person is 10. None of the 10 samples is identical to any other. They vary in position, rotation, scale and expression. The changes in orientation have been accomplished by rotating the person a maximum of 20° in the same plane; also each person has changed his/her facial expression in each of 10 samples (open/closed eye, smiling/not smiling). The changes in scale have been achieved by changing the distance between the person and the video camera. For some individuals, the images were taken at different time, varying facial details (glasses/no glasses). In addition, the images have been manually cropped and re-scaled to a resolution of 112 × 92, 8-bit grey levels. Thumbnails of all the images are shown in Figure 5.
All experiments were executed with one randomly selected training images and the other nine test images per person for a total of 40 training and 360 test images. It should be noted that there is no overlap between the training and test images. Each case was repeated 30 times by randomly choosing different training and testing sets and the average recognition rate is used to evaluate the classification performance. The face recognition experiments were implemented in C++ programming language on an Intel Core 2 2.34 GHz computer with 2G RAM running on Window Vista. It took nearly 33 seconds ($N_{th}=1$) to 40 seconds ($N_{th}=20$) training the PCA-IN, respectively. However, once the whole PCA-IN (consisting 40 immune network classifiers) is determined, it takes less than 20 ms to recognize a face. A simulation window of face recognition, its setting parameters, and the results are shown in Figure 6. This figure indicates that one randomly selected image (red block) is selected for training, while the testing image is identical to the image with yellow block. It demonstrates that the PCA-IN correctly recognizes the image as the 19th person.
First of all, PCA is utilized to obtain eigenvalues and eigenvectors of the face images. Afterward, the randomly selected single training sample is input into the immune networks which are optimized using GAs. The population size of GAs is 200 while the stopping criterion is 1000 generations. The operators utilized are Roulette Wheel selection, arithmetic crossover and Gaussian mutation with crossover and mutation rate of 0.8 and 0.05, respectively. The performances of the proposed method are evaluated by varying the number of antibodies of each immune classifier ($N_{Ab}$), eigenfaces ($N_{eig}$) and the weighting values ($w_1$ and $w_2$). Each experiment was repeated for 30 times and the "average recognition rate" (ARR) is illustrated in Table 2 and Table 3 for $w_1=0.5$ and $w_1=1.0$, respectively. In addition, Figure 7 indicates the average recognition rates with respect to $N_{Ab}$ and the weighting values. Clearly, ARR is nearly increased proportional to the number of antibodies adopted in the immune network classifiers. Additionally, it seems that ARR increases exponentially when $N_{Ab}$ is between 1 and 10. On the contrary, there is no much improvement of ARR if $N_{Ab}$ is above 10. In addition, it seems that the number of eigenfaces $N_{eig}$ has not consistent effect on the ARR values. Moreover, it shows that the proposed PCA-IN achieves bigger ARR when $w_1$ equals to 1.0. In such situation, each immune classifier can be trained simply utilizing its corresponding person images and thus the training of the 40 immune classifiers can be decoupled. In other words, immune network classifier for any new person image can be trained separately and involved in the immune networks without affect the existing immune network classifiers.

Table 2. ARR(%) VS $N_{Ab}$ and $N_{eig}$ ($w_1=0.5$; $w_2=0.5$)

<table>
<thead>
<tr>
<th>$N_{Ab}$</th>
<th>$N_{eig}=5$</th>
<th>$N_{eig}=10$</th>
<th>$N_{eig}=20$</th>
<th>$N_{eig}=30$</th>
<th>$N_{eig}=40$</th>
<th>ARR+SD (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>78.18</td>
<td>77.55</td>
<td>77.78</td>
<td>78.24</td>
<td>78.13</td>
<td>77.98±0.30</td>
</tr>
<tr>
<td>2</td>
<td>88.48</td>
<td>88.14</td>
<td>88.46</td>
<td>88.19</td>
<td>88.50</td>
<td>88.35±0.17</td>
</tr>
<tr>
<td>3</td>
<td>92.57</td>
<td>92.64</td>
<td>93.01</td>
<td>92.29</td>
<td>92.31</td>
<td>92.56±0.29</td>
</tr>
<tr>
<td>4</td>
<td>94.17</td>
<td>94.09</td>
<td>94.51</td>
<td>94.60</td>
<td>94.67</td>
<td>94.41±0.26</td>
</tr>
<tr>
<td>5</td>
<td>95.77</td>
<td>95.80</td>
<td>95.62</td>
<td>95.56</td>
<td>95.84</td>
<td>95.72±0.12</td>
</tr>
<tr>
<td>6</td>
<td>96.41</td>
<td>96.64</td>
<td>96.34</td>
<td>96.19</td>
<td>96.58</td>
<td>96.43±0.18</td>
</tr>
<tr>
<td>7</td>
<td>96.96</td>
<td>97.16</td>
<td>96.59</td>
<td>96.90</td>
<td>96.81</td>
<td>96.88±0.21</td>
</tr>
<tr>
<td>8</td>
<td>97.44</td>
<td>97.41</td>
<td>97.00</td>
<td>97.20</td>
<td>97.25</td>
<td>97.26±0.18</td>
</tr>
<tr>
<td>9</td>
<td>97.64</td>
<td>97.72</td>
<td>97.44</td>
<td>97.52</td>
<td>97.33</td>
<td>97.53±0.16</td>
</tr>
<tr>
<td>10</td>
<td>97.90</td>
<td>97.80</td>
<td>97.75</td>
<td>97.82</td>
<td>97.61</td>
<td>97.78±0.11</td>
</tr>
<tr>
<td>15</td>
<td>98.53</td>
<td>98.07</td>
<td>98.55</td>
<td>98.30</td>
<td>98.49</td>
<td>98.39±0.20</td>
</tr>
<tr>
<td>20</td>
<td>98.61</td>
<td>98.75</td>
<td>98.70</td>
<td>98.80</td>
<td>98.77</td>
<td>98.73±0.07</td>
</tr>
</tbody>
</table>
Table 3. ARR(%) VS $N_{Ab}$ and $N_{eig}$ ($w_1=1.0$; $w_2=0.0$)

<table>
<thead>
<tr>
<th>$N_{Ab}$</th>
<th>$N_{eig}=5$</th>
<th>$N_{eig}=10$</th>
<th>$N_{eig}=20$</th>
<th>$N_{eig}=30$</th>
<th>$N_{eig}=40$</th>
<th>ARR+SD (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>85.48</td>
<td>86.67</td>
<td>85.91</td>
<td>85.87</td>
<td>86.67</td>
<td>86.12±0.53</td>
</tr>
<tr>
<td>2</td>
<td>93.34</td>
<td>94.13</td>
<td>94.14</td>
<td>93.49</td>
<td>93.59</td>
<td>93.74±0.31</td>
</tr>
<tr>
<td>3</td>
<td>96.61</td>
<td>96.58</td>
<td>96.69</td>
<td>96.53</td>
<td>95.86</td>
<td>96.45±0.34</td>
</tr>
<tr>
<td>4</td>
<td>97.66</td>
<td>97.78</td>
<td>97.63</td>
<td>97.67</td>
<td>97.57</td>
<td>97.66±0.08</td>
</tr>
<tr>
<td>5</td>
<td>98.01</td>
<td>98.41</td>
<td>98.13</td>
<td>97.95</td>
<td>98.41</td>
<td>98.18±0.22</td>
</tr>
<tr>
<td>6</td>
<td>98.79</td>
<td>98.82</td>
<td>98.51</td>
<td>98.46</td>
<td>98.61</td>
<td>98.64±0.16</td>
</tr>
<tr>
<td>7</td>
<td>98.96</td>
<td>98.84</td>
<td>98.73</td>
<td>98.72</td>
<td>98.91</td>
<td>98.83±0.11</td>
</tr>
<tr>
<td>8</td>
<td>98.98</td>
<td>99.14</td>
<td>98.96</td>
<td>98.93</td>
<td>99.12</td>
<td>99.03±0.10</td>
</tr>
<tr>
<td>9</td>
<td>99.02</td>
<td>99.15</td>
<td>99.09</td>
<td>99.18</td>
<td>99.16</td>
<td>99.12±0.07</td>
</tr>
<tr>
<td>10</td>
<td>99.26</td>
<td>99.33</td>
<td>99.12</td>
<td>99.35</td>
<td>99.31</td>
<td>99.27±0.09</td>
</tr>
<tr>
<td>15</td>
<td>99.44</td>
<td>99.47</td>
<td>99.59</td>
<td>99.50</td>
<td>99.61</td>
<td>99.52±0.07</td>
</tr>
<tr>
<td>20</td>
<td>99.64</td>
<td>99.68</td>
<td>99.69</td>
<td>99.78</td>
<td>99.72</td>
<td>99.70±0.05</td>
</tr>
</tbody>
</table>

Figure 7. Average recognition rate vs. number of antibodies

Finally, the best performance (99.70%) of the proposed immune network classifiers was compared with the results reported in [8-16, 22-43] as shown in Table 4. It should be noted that some researchers selected the single training image per person randomly while the others adopt fixed training images. Therefore, the recognition rates listed in Table 4 are either average recognition rate or top recognition rate. Obviously, the proposed PCA-IN method outperformed all other methods.

Table 4. Comparison of ARR of PCA-IN with some other methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFLDA [8]</td>
<td>74.28%</td>
</tr>
<tr>
<td>S2DLDA [9]</td>
<td>76.17%</td>
</tr>
<tr>
<td>SVD [10]</td>
<td>69.56%</td>
</tr>
<tr>
<td>Enhancement method based on WT [11]</td>
<td>89.72%</td>
</tr>
<tr>
<td>Projection map and SVD [12]</td>
<td>88.61%</td>
</tr>
<tr>
<td>2DPCA [13]</td>
<td>76.70%</td>
</tr>
<tr>
<td>RBF SVM [14]</td>
<td>84.50%</td>
</tr>
<tr>
<td>DDCT and 2PCA [15]</td>
<td>76.25%</td>
</tr>
<tr>
<td>Modular Weighted (2D)²PCA [16]</td>
<td>72.22%</td>
</tr>
<tr>
<td>FLDA [22]</td>
<td>76.39%</td>
</tr>
<tr>
<td>2D(PC)²A [23]</td>
<td>60.00%</td>
</tr>
<tr>
<td>2DPCA with virtual views [24]</td>
<td>82.10%</td>
</tr>
<tr>
<td>Block based DT-CWT(μ,σ) [25]</td>
<td>78.44%</td>
</tr>
</tbody>
</table>
6. Conclusions

This paper presented PCA-IN architecture for face recognition with one training sample per person. The performance of the PCA-IN was evaluated using the ORL face database. The results show that the proposed method obtains good recognition rate if only the corresponding person trained images are considered in the immune classifier (i.e. $w_1=1.0$). This result greatly increases the practical application of the proposed method in real life. Moreover, the proposed immune networks provide best recognition rate compared with all the other developed technologies.

7. Acknowledgements

The authors would like to acknowledge the National Science Council, Taiwan, R.O.C., for making this work possible with grants NSC93-2213-E-036-016 and NSC94-2213-E-036-002.

8. References


<table>
<thead>
<tr>
<th>Method</th>
<th>Recognition Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-AFMT+N_OFMM [26]</td>
<td>74.00%</td>
</tr>
<tr>
<td>m-MSD+SVD [27]</td>
<td>78.50%</td>
</tr>
<tr>
<td>DCT-PCA [28]</td>
<td>75.56%</td>
</tr>
<tr>
<td>Wavelet Transform [29]</td>
<td>73.89%</td>
</tr>
<tr>
<td>Variance weight LBP partition in TCFE [30]</td>
<td>80.25%</td>
</tr>
<tr>
<td>Hybrid Fourier-AFMT [31]</td>
<td>80.00%</td>
</tr>
<tr>
<td>Local DT-CWT parallelogram Block [32]</td>
<td>81.78%</td>
</tr>
<tr>
<td>Pseudo-Fisherface Method [33]</td>
<td>72.40%</td>
</tr>
<tr>
<td>Local Binary Patter approach [34]</td>
<td>86.77%</td>
</tr>
<tr>
<td>Sample augment and improved 2DPCA [35]</td>
<td>76.39%</td>
</tr>
<tr>
<td>Gabor filter + Enhanced majority voting [36]</td>
<td>74.44%</td>
</tr>
<tr>
<td>MR_2DLDA [37]</td>
<td>83.89%</td>
</tr>
<tr>
<td>Projection map and Fourier Transform (rank10) [38]</td>
<td>95.85%</td>
</tr>
<tr>
<td>Spectroface + uniform eigen-space SVD (rank 3) [39]</td>
<td>89.72%</td>
</tr>
<tr>
<td>Contourlet based NDNFLA [40]</td>
<td>71.00%</td>
</tr>
<tr>
<td>Fusing 2DKPCA and 2D(PC)^2A [41]</td>
<td>67.00%</td>
</tr>
<tr>
<td>S2DLDA [42]</td>
<td>76.17%</td>
</tr>
<tr>
<td>Kernel partial-least-squares regression [43]</td>
<td>75.90%</td>
</tr>
<tr>
<td>Proposed PCA-IN</td>
<td>99.70%</td>
</tr>
</tbody>
</table>


