SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 9, No. 1, February 2012, 121-130

UDK: 004.932.72'1 DOI: 10.2298/SJEE1201121S

Face Recognition Using Eigenface Approach*

Marijeta Slavković[1], Dubravka Jevtić[1]

Abstract: In this article, a face recognition system using the Principal Component Analysis (PCA) algorithm was implemented. The algorithm is based on an eigenfaces approach which represents a PCA method in which a small set of significant features are used to describe the variation between face images. Experimental results for different numbers of eigenfaces are shown to verify the viability of the proposed method.

Keywords: PCA, Face recognition, Eigenface.

1 Introduction

The human face is an extremely complex and dynamic structure with characteristics that can quickly and significantly change with time. It is the primary focus of attention in social relationships and plays a major role in the transmission of identity and emotions. Therefore, face recognition is applied in many important areas such as security systems, identification of criminals, verification of credit cards and so on. Unfortunately, many face features make development of facial recognition systems difficult.

This problem is solved by the method called Principal Component Analysis. PCA is a projection technique that finds a set of projection vectors designed such that the projected data retains the most information about the original data. The most representative vectors are eigenvectors corresponding to highest eigenvalues of the covariance matrix. This method reduces the dimensionality of data space by projecting data from M-dimensional space to P-dimensional space, where $P \ll M$.

2 Face Recognition

One of the simplest and most effective PCA approaches used in face recognition systems is the so-called eigenface approach. This approach transforms faces into a small set of essential characteristics, eigenfaces, which are the main components of the initial set of learning images (training set).

[1]Innovation Center of School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia; E-mails: marijeta.slavkovic@gmail.com; dubravka.jevtic@gmail.com
*Award for the best paper presented in Section *Electrical Circuits*, at Conference ETRAN 2011, June 6 – 9, Banja Vrućica – Teslić, Bosnia and Herzegovina.

Recognition is done by projecting a new image in the eigenface subspace, after which the person is classified by comparing its position in eigenface space with the position of known individuals [1].

The advantage of this approach over other face recognition systems is in its simplicity, speed and insensitivity to small or gradual changes on the face. The problem is limited to files that can be used to recognize the face. Namely, the images must be vertical frontal views of human faces.

3 PCA Approach to Face Recognition

Principal component analysis transforms a set of data obtained from possibly correlated variables into a set of values of uncorrelated variables called principal components. The number of components can be less than or equal to the number of original variables. The first principal component has the highest possible variance, and each of the succeeding component has the highest possible variance under the restriction that it has to be orthogonal to the previous component. We want to find the principal components, in this case eigenvectors of the covariance matrix of facial images.

The first thing we need to do is to form a training data set. 2D image I_i can be represented as a 1D vector by concatenating rows [2]. Image is transformed into a vector of length $N = mn$.

$$\boldsymbol{I} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}_{mxn} \xrightarrow{\mathit{CONCATENATION}} \begin{bmatrix} x_{11} \\ \vdots \\ x_{1n} \\ \vdots \\ x_{2n} \\ \vdots \\ x_{mn} \end{bmatrix}_{1xN} = \boldsymbol{x}\,.$$

Let M such vectors $\boldsymbol{x}_i$ ($i = 1, 2, ..., M$) of length N form a matrix of learning images, $\boldsymbol{X}$. To ensure that the first principal component describes the direction of maximum variance, it is necessary to center the matrix. First we determine the vector of mean values $\boldsymbol{\Psi}$, and then subtract that vector from each image vector.

$$\boldsymbol{\Psi} = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{x}_i\,, \tag{1}$$

$$\phi_i = \boldsymbol{x}_i - \boldsymbol{\Psi}\,. \tag{2}$$

Averaged vectors are arranged to form a new training matrix (size $N\times M$); $\boldsymbol{A} = (\phi_1, \phi_2, \ldots, \phi_M)$.

The next step is to calculate the covariance matrix $\boldsymbol{C}$, and find its eigenvectors $\boldsymbol{e}_i$ and eigenvalues $\boldsymbol{\lambda}_i$:

$$\boldsymbol{C} = \frac{1}{M}\sum_{n=1}^{M} \varphi_n \varphi_n^{\ T} = \boldsymbol{A}\boldsymbol{A}^T , \tag{3}$$

$$\boldsymbol{C}\boldsymbol{e}_i = \boldsymbol{\lambda}_i \boldsymbol{e}_i . \tag{4}$$

Covariance matrix $\boldsymbol{C}$ has dimensions $N\times N$. From that we get N egeinvalues and eigenvectors. For an image size of 128×128, we would have to calculate the matrix of dimensions 16.384×16.384 and find 16.384 eigenvectors. It is not very effective since we do not need most of these vectors. Rank of covariance matrix is limited by the number of images in learning set — if we have M images, we will have M–1 eigenvectors corresponding to non-zero eigenvalues. One of the theorems in linear algebra states that the eigenvectors e_i and eigenvalues λ_i can be obtained by finding eigenvectors and eigenvalues of matrix $\boldsymbol{C}_1 = \boldsymbol{A}^T\boldsymbol{A}$ (dimensions $M\times M$) [3]. If $\boldsymbol{v}_i$ and $\boldsymbol{\mu}_i$ are eigenvectors and eigenvalues of matrix $\boldsymbol{A}^T\boldsymbol{A}$, then:

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{v}_i = \boldsymbol{\mu}_i \boldsymbol{v}_i . \tag{5}$$

Multiplying both sides of equation (5) with $\boldsymbol{A}$ from the left, we get:

$$\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{v}_i = \boldsymbol{A}\boldsymbol{\mu}_i \boldsymbol{v}_i ,$$

$$\boldsymbol{A}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{v}_i) = \boldsymbol{\mu}_i (\boldsymbol{A}\boldsymbol{v}_i) ,$$

$$\boldsymbol{C}(\boldsymbol{A}\boldsymbol{v}_i) = \boldsymbol{\mu}_i (\boldsymbol{A}\boldsymbol{v}_i) . \tag{6}$$

Comparing equations (4) and (6) we can conclude that the first M–1 eigenvectors $\boldsymbol{e}_i$ and eigenvalues $\boldsymbol{\lambda}_i$ of matrix $\boldsymbol{C}$ are given by $\boldsymbol{A}\boldsymbol{v}_i$ and $\boldsymbol{\mu}_i$, respectively. Eigenvector associated with the highest eigenvalue reflects the highest variance, and the one associated with the lowest eigenvalue, the smallest variance. Eigenvalues decrease exponentially so that about 90% of the total variance is contained in the first 5% to 10% eigenvectors [3]. Therefore, the vectors should be sorted by eigenvalues so that the first vector corresponds to the highest eigenvalue. These vectors are then normalized. They form the new matrix $\boldsymbol{E}$ so that each vector $\boldsymbol{e}_i$ is a column vector. The dimensions of this matrix are $N\times D$, where D represents the desired number of eigenvectors. It is used for projection of data matrix $\boldsymbol{A}$ and calculatation of $\boldsymbol{y}_i$ vectors of matrix $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M)$:

$$\boldsymbol{Y} = \boldsymbol{E}^T\boldsymbol{A} . \tag{7}$$

Each original image can be reconstructed by adding mean image $\boldsymbol{\Psi}$ to the weighted summation of all vectors $\boldsymbol{e}_i$.

The last step is the recognition of faces. Image of the person we want to find in training set is transformed into a vector $\boldsymbol{P}$, reduced by the mean value $\boldsymbol{\Psi}$ and projected with a matrix of eigenvectors (eigenfaces):

$$\boldsymbol{\omega} = \boldsymbol{E}^T(\boldsymbol{P} - \boldsymbol{\Psi}). \tag{8}$$

Classification is done by determining the distance, $\boldsymbol{\varepsilon}_i$, between $\boldsymbol{\omega}$ and each vector y_i of matrix $\boldsymbol{Y}$. The most common is the Euclidean distance, but other measures may be used. This paper presents the results for the Euclidean and Manhattan distance. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are two vectors of length $\boldsymbol{D}$, the distance between them is determined as follows [4]:

1) Manhattan distance:

$$d(\boldsymbol{A},\boldsymbol{B}) = \sum_{i=1}^{D} |a_i - b_i|; \tag{9}$$

2) Euclidean distance:

$$d(\boldsymbol{A},\boldsymbol{B}) = \sqrt{\sum_{i=1}^{D} (a_i - b_i)^2} = \|\boldsymbol{A} - \boldsymbol{B}\|. \tag{10}$$

If the minimum distance between test face and training faces is higher than a threshold θ, the test face is considered to be unknown, otherwise it is known and belongs to the person $s = \arg\min_i [\varepsilon_i]$ [4].

Why is setting up a threshold important? The program requires a minimum distance between the test image and images from the training base. Even if the person is not in the database, the face would be recognized. It is therefore necessary to set a threshold that will allow us to determine whether a person is in the database. There is no formula for determining the threshold. The most common way is to first calculate the minimum distance of each image from the training base from the other images and place that distance in a vector *rast*. Threshold is taken as 0.8 times of the maximum value of vector *rast* [5]:

$$\theta = 0.8 * \max(rast). \tag{11}$$

4 Experimental Results

The experiment was conducted using ORL database of faces [6]. The training database contains 190 images of 38 persons (5 images per each person), a test database has 40 images of different individuals (38 known and 2 unknown). All photos have dimensions 92×112 and a dark homogeneous background and the subject is photographed in an upright, frontal position. All images are grayscale (intensity levels of gray are taken as image features).

Example of images from the training base is given in Fig. 1. Fig. 2 shows all 190 eigenvalues. Each eigenvalue corresponds to a single eigenvector and tells us how much images from training bases vary from the mean image in that direction. It can be seen that about 10% of vectors have significant eigenvalues, while those for the remaining vectors are approximately equal to zero. We do not have to take into account eigenvectors that correspond to small eigenvalues because they do not carry important information about the image.

Fig. 1 – *Pictures from the training base.*

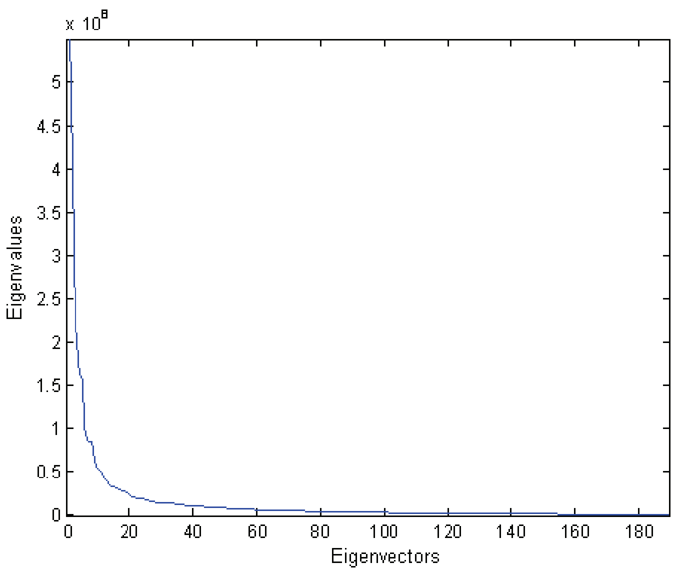

Fig. 2 – *Eigenvalues.*

In Figs. 3 and 4 first three and last three eigenfaces are shown, respectively. While the figures in Fig. 3 resembles the faces, those in Fig. 4 do not carry important information about the images from the training base.

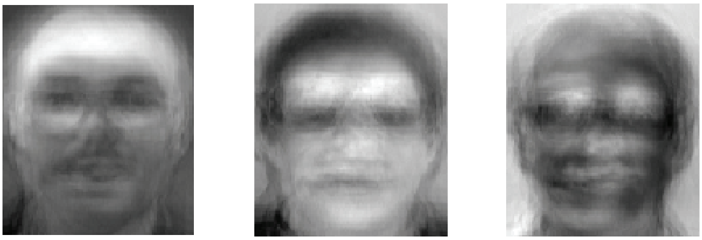

Fig. 3 – *First three eigenfaces.*

Fig. 4 – *Last three eigenfaces.*

Table 1 shows that the proposed algorithm gives the same results regardless of whether we used only the first 20 or all 190 eigenvectors. Recognition rate is higher for Manhattan than for Euclidean distance if we use 5 or 10 eigenvectors, while for 20 eigenvectors it is the same in both cases.

In practice, only a few images per person are available for the training base, so it is important to note the effect of a number of images per subject on the rate of recognition. The corresponding recognition rates for different number of training images per subject are given in **Table 2**. For comparison, we selected the first 20 principal components in each case.

Table 1

The result of face recognition using eigenface.

Number of Principal Components	RECOGNITION RATE	
	Euclidean distance	Manhattan distance
5	77.5 %	80 %
10	92.5 %	95 %
20	97.5 %	97.5 %
190	97.5 %	97.5 %

Table 2
Recognition rate for different number of images per person.

RECOGNITION RATE					
No. of images per person	1	2	3	4	5
Euclidean distance	70 %	87.5 %	87.5 %	92.5 %	97.5 %
Manhattan distance	67.5 %	77.5 %	82.5 %	92.5 %	97.5 %

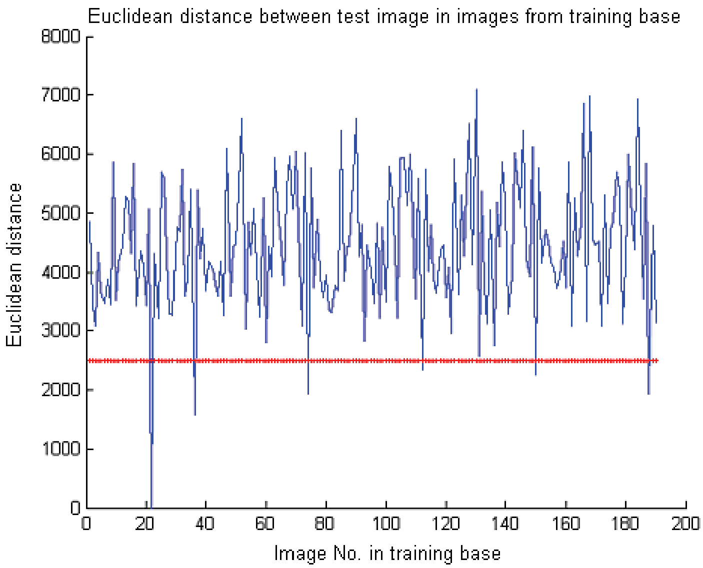

Fig. 5 – *Euclidean distance between the test image and images from the database.*

Fig. 6 – *Test image and recognized image from the training base.*

To recognize the face we calculate the distance of test image from each image from the training base. Minimum distance shows us which image from the database matches the test image best. Fig. 5 shows the Euclidean distance between the test image and all 190 images from database. The distance is minimum for the image No. 21 and is equal to zero. This means that the test image completely matches the picture 21 from the training base (Fig. 6).

The second test was done for the person who was also present in the database but has a different facial expression. The test image has a minimum Euclidean distance which is less than the threshold (Fig. 7), so it is a known face (Fig. 8).

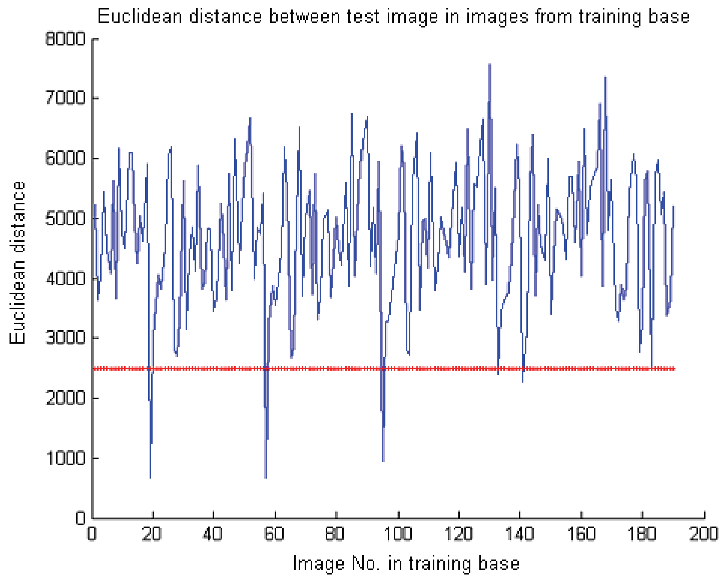

Fig. 7 – *Euclidean distance between the test image and images from the database.*

Fig. 8 – *Test image and recognized image from the training base.*

The minimum Euclidean distance for the third test image (Fig. 9) is greater than the threshold, therefore the test image represents an unknown face (Fig. 10).

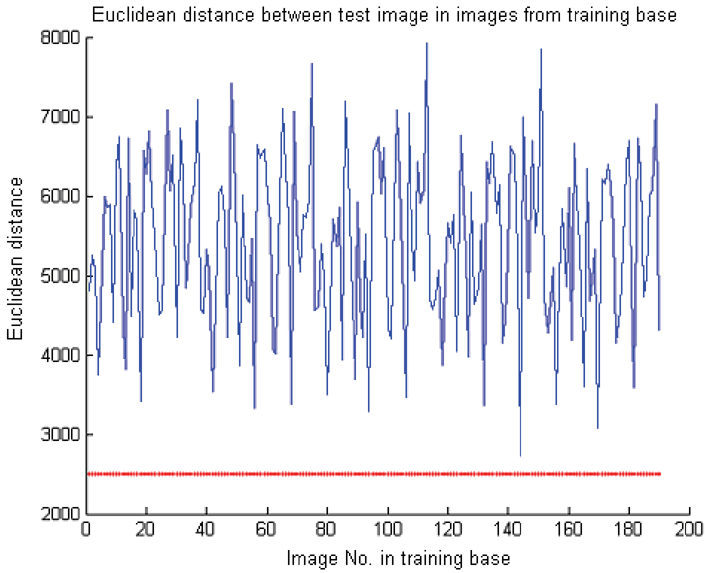

Fig. 9 – *Euclidean distance between the test image and images from the database.*

Fig. 10 – *Image of unknown person.*

4 Conclusion

Face recognition method using eigenfaces is proposed. We used database of face images which contains 190 images of 38 different persons (5 images per person). From the results, it can be concluded that, for recognition, it is sufficient to take about 10% eigenfaces with the highest eigenvalues. It is also clear that the recognition rate increases with the number of training images per person. It is obvious that if the minimum distance between the test image and other images is zero, the test image entirely matches the image from the training base. If the distance is greater than zero but less than a certain threshold, it is a known person with other facial expression, otherwise it is an unknown person.

5 Acknowledgement

The research described in this paper were partially funded by the Ministry of Education and Science of Serbia through the project of Integrated and interdisciplinary research program, number 44009.

6 References

[1] M. Turk, A. Pentland: Face Recognition using Eigenfaces, Conference on Computer Vision and Pattern Recognition, 3 – 6 June 1991, Maui, HI , USA, pp. 586 – 591.

[2] V. Perlibakas: Face Recognition using Principal Component Analysis and Wavelet Packet Decomposition, Informatica, Vol. 15, No. 2, 2004, pp. 243 – 250.

[3] K. Kim: Face Recognition using Principal Component Analysis, National Institute of Technology, Rourkela, India, 2008.

[4] V. Perlibakas: Distance Measures for PCA-based Face Recognition, Pattern Recognition Letters, Vol. 25, No. 6, April 2004, pp. 711 – 724.

[5] S. Gupta , O.P. Sahoo, A. Goel, R. Gupta: A New Optimized Approach to Face Recognition using EigenFaces, Global Journal of Computer Science and Technology, Vol. 10, No. 1, April 2010, pp. 15 – 17.

[6] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.