ISBN 978-981-11-3671-9
Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering (WCSE 2017)
Beijing, 25-27 June, 2017, pp. 272-277
doi: 10.18178/wcse.2017.06.047

Automatic Face Recognition using Principal Component Analysis and Neural Network

Mirabeau Nguela [1], Zenghui Wang[1], Yanxia Sun[2+]

[1] Department of Electrical and Mining Engineering, University of South Africa, Florida 1710, South Africa

[2] Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg 2006, South Africa

Abstract. Automatic face recognition is a very important research area in computer science since it has been widely used in security systems. It has drawn a lot of attention in the recent ten years from the scientific communities with the aim to provide highly intelligent human-machine interaction with high performance. This paper proposes an automatic face recognition method that encompasses a reduction of significant variable features using the principal components analysis and classification method through Neural Network. The experimental results obtained show an improvement in term of recognition rate compared with the existing method.

Keywords: Principal component analysis, neural network, eigenvalues and eigenvectors, covariance matrix, feed forward network

1. Introduction

Our interaction between human has a mains point of focus: the face of our interlocutor. Human has outstanding abilities when it comes to recognize and differentiate objects and particularly others humans through their faces although there are some occlusions or changing conditions such as age, hairstyle, beard, glasses, and so on. Researchers throughout history have tried to emulate these humans' like qualities by developing a computational model. The relevance of research is wide in its multidisciplinary applications. The application field covers not only computer vision and pattern recognition for criminal identification [1], security and law enforcement but also in computer graphics, control systems, image processing and even psychology. The face recognition has a long history.

The field of psychology in the 1950's has been the first to experience the earliest research work on face recognition [2] where issues like interpretation of emotion; face expression, amongst others were studied. Fischler and Elschanger tried to measure similar features automatically. Their algorithm used local template matching and a global measure of fit to find and measure facial features [3]. The extraction of features of faces such as mouth, ears nose eyes, chin position and the inter-relation between them were the main focus [4] however such methods were very shallow for expressing features of face images. Some algorithms to extract the features of faces have been proposed, and one of the most important methods is Principal Component Analysis (PCA) [5,6,9,10]. To further improve the performance this paper proposed an automatic face recognition using principal component analysis and neural network.

2. Principal Component Analysis (PCA)

[+] Corresponding author. Tel.: + +27115592213.
E-mail address: sunyanxia@gmail.com.

The principal component analysis method is used to perform dimensionality reduction while preserving as much of the variance in the high-dimensional space as possible. Principal component analysis [5] uses factorization to transform data according to its statistical properties.

The image data has a size of $m \times n$ pixels and represented by

$X = \{x_1, x_2, ..., x_m\}$

where x_m is a set of m vectors and where each vector x_i has n elements concatenated (for example, Fig1), for each image; the element is in fact the gray level value of each image. First we create a data matrix, consisting of pixel values in columns.

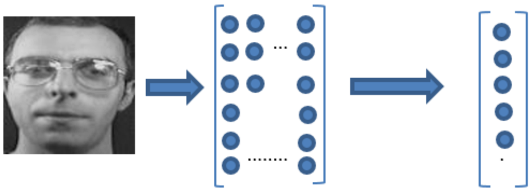

Fig.1: Sample face image of ATT & T data base

The image data must be centered by subtracting the mean image from each of the training images. This is an essential preprocessing step for the principal component algorithm.

$$S_T = \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T \quad (1)$$

The centered data matrix (X') is multiplied by its transpose to create a covariance matrix

$$C = XX^T \quad (2)$$

where the eigenvalue analysis is applied. The eigenvectors selected account for the basis vectors of the new space defined by the reduced sample images. In order to project the images to the new space, the dot product between the image and each of the eigenvectors is computed. The final dimension of the reduced feature vector is equal to the chosen number of eigenvectors; each image in column-vector form is multiplied to the transpose of X=X'

3. Neural Network

Fig. 2 represents one neuron within the network; it is a simple unit that process all the input and delivers the output to be passed to the next input, that is

Output $a = f(p_1w_1 + p_2w_2 + p_3w_3 + Bias) = f(\sum p_iw_i + Bias)$

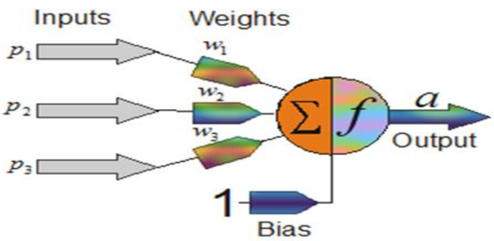

Fig.2: One neuron connection

In this study, a two-layer real-output Network is used. This network computes a function f from to R^n to R of the form:

$$f(X) = \sum_{i=1}^{k} w_i \sigma(v_i x + v_{i,0}) + w_0 \quad (3)$$

where x ϵ R^n is the input vector, $w_i \in R$ (i = 0,1,2..., k) is the output weight, $v_i \in R^n$ and $v_{i,0}$ (i = 0,1,2..., k) are the input weights, and $\sigma : R \to R$.

Using the neural network pattern recognition tool GUI, nprtool (Matlab), we are able to proceed with the training and testing and the result were obtained. In order to optimize network performance, we can tune the value of the weights and bias of the network during the training as defined by the network performance function net.performFcn. The performance default function for feedforward networks is mean square error mse (4): the average squared error between the network outputs O and the target outputs T. The actual output of the network is compared to expected output for that particular input. This results in an error value. The connection weights in the network are gradually adjusted, working backwards from the output layer, through the hidden layer, and to the input layer, until the correct output is produced.

$$F = mse = \frac{1}{N}\sum_{i=1}^{N} e_i^2 = \frac{1}{N}\sum_{i=1}^{N}(T_i - O_i)^2 \tag{4}$$

4. Neural Network based PCA

To improve the performance of PCA, a neural network based PCA is proposed and its steps are

1) Collect data

A set features vectors for training the network is assembled. Each vectors consists of a value of PCA (which represents the input vector into the network) and the corresponding solution (which represents the target output from the network). The input data is entered into the network via the input layer. Each neuron in the network processes the input data with the resultant values steadily till the optimum result is given.

The actual output of the network is compared to the target or expected output for that particular input.

2) Create the network

The number of hidden layer is determined by trial and error during training process

3) Configure the network

4) Initialize the weights and biases

5) Train the network

The connection weights in the network are gradually adjusted, working backwards from the output layer, through the hidden layer, and to the input layer, until the correct output is produced.

6) Validate the network (post-training analysis)

Fine tuning the weights in this way has the effect of teaching the network how to produce the correct output for a particular input, i.e. the network learns.

7) Apply the network.

And the corresponding flow chart is shown in Fig.5. The blocks 1-4 of the flowchart are corresponding to the step 1). The blocks at level 5 are corresponding to the steps 2)-6); and the blocks 6-8 are corresponding to the step 7).

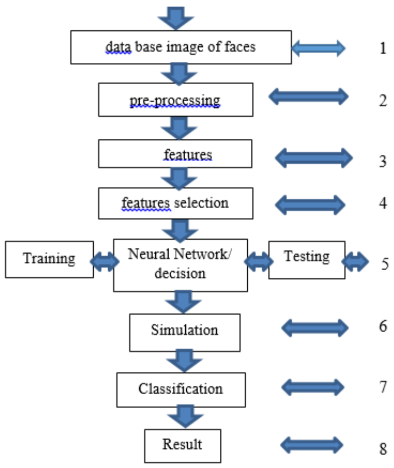

Fig.3: Flowchart Diagram

5. Experiments and Analyses

The basis of the beginning of the work was to have substantial amount of faces image. Here, two sets of faces are used to test the proposed method. One set is from the Cambridge University Engineering Department (www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.). The overall images of AT&T is m x n pixels and were taken against a dark homogeneous background (that make it avoid a pre-processing step) with the subjects in an upright, frontal position (with tolerance for some side movement). The data base consists of ten different images. Another set of images was taken at different times, varying the lighting, facial expressions face image used for study (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). Some of the face images have been shown in Fig. 4 and Fig. 5. The neural network structure chosen is shown in Fig. 6, and the weights and bias of the neural network will be trained based on the default method.


Fig.4: Partial faces of the AT&T Database Images [7]


Fig.5: Some images from the http://www.fael.edu.br (Data base)[8]

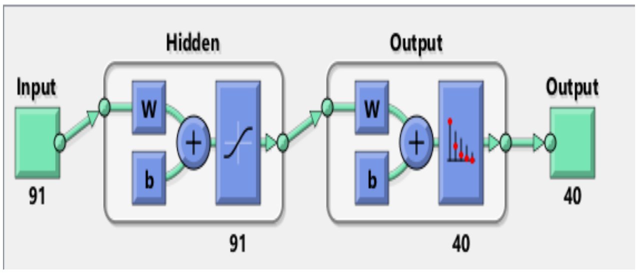

Fig. 6: Neural Network connection diagram

The total number of image object is divided into two sets. One is used for the training and the others is used to evaluate the performance of the network once the optimum training is performed. The experimental results are shown in the Table 1 and table 2 and Fig. 7.

Table 1, which corresponds to the result of experiment using the first database (www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.), could be described as follows: on the first column there are indices used to increase or decrease the amount of PCA extracted from the (1) and (2) , then the data is divided into two sets, training sets and testing sets . The operation using the two sets are performed; the result is shown in the following column with the amount of images used. The same is done for the second database (http://www.fael.edu.br data base). Also the number of features vectors provides a sparse result however without linearly improving it. As an example for an index of 0.85, we achieved 93.3% recognition rate and 92.2% for 0.9 value index. Note: the first row (the simulation results) of each components index did not use neural network and the second row used the proposed method for Tables 1 and 2.

In Fig. 7, there are three blocks of colors: the confusion matrix shows the percentage of correct and incorrect classifications. Correct classifications are the green squares on the matrices diagonal. Incorrect classifications form the red squares. The blue square is the overall performance of correct and incorrect data. The dark square gives performance of correct and misclassified data of each class.

Table 1: Result obtained for different value specified using the Att&t data base

Coeff of Principal Components (used to varied the amount of value components) index	features vectors Amount	Performance respond computation times		percentage of recognition success With unknown images	Total faces: 400 Numbers of image per classes :10	
		Training(cs)	Testing MSE (cs)		Training	Testing
0.85	57	3.7×10^{-7}	8.22×10^{-2}	90.8%	6x30	4x30
		9.34×10^{-7}	6.94×10^{-2}	93.3%	7x30	3x30

0.86	62	4.85×10^{-7}	8.29×10^{-2}	89.9%	6x30	4x30
		6.72×10^{-7}	6.87×10^{-2}	91.1%	7x30	3x30
0.87	68	6.11×10^{-7}	9.74×10^{-2}	89.5%	6x30	4x30
		5.14×10^{-7}	6.77×10^{-2}	92.2%	7x30	3x30
0.88	75	6.85×10^{-7}	7.63×10^{-2}	90.8%	6x30	4x30
		5.92×10^{-7}	8.83×10^{-2}	90%	7x30	3x30
0.9	90	3.53×10^{-7}	1.51×10^{-2}	88.9%	6x30	4x30
		5.13×10^{-7}	1.06×10^{-2}	92,2%	7x30	3x30

Table 2: Result obtained for different value specified using the http://www.fael.edu.br data base

Coeff of Principal					Total faces: 1400	
Components (used to varied the amount of value components) index	features vectors Amount	Performance respond computation times		percentage of recognition success with unknown images	Numbers of image per classes :14	
		Training(cs)	Testing MSE (cs)		Training	Testing
0.85	33	8.80×10^{-6}	15.19×10^{-2}	91.8%	8x70	6x70
		15.44×10^{-6}	9.86×10^{-2}	92.3%	10x70	4x70
0.86	37	9.75×10^{-6}	13.34×10^{-2}	91.9%	8x70	6x70
		11.82×10^{-6}	12.90×10^{-2}	92.4%	10x70	4x70
0.87	42	11.21×10^{-6}	16.80×10^{-2}	90.5%	8x70	6x70
		10.19×10^{-6}	13.85×10^{-2}	91.5%	10x70	4x70
0.88	47	12.88×10^{-6}	15.69×10^{-2}	90.8%	8x70	6x70
		11.82×10^{-6}	15.90×10^{-2}	91,2%	10x70	4x30
0.9	61	7.63×10^{-6}	3.65×10^{-2}	89.35.9%	8x70	6x70
		10.17×10^{-6}	3.06×10^{-2}	93,2%	10x70	4x70

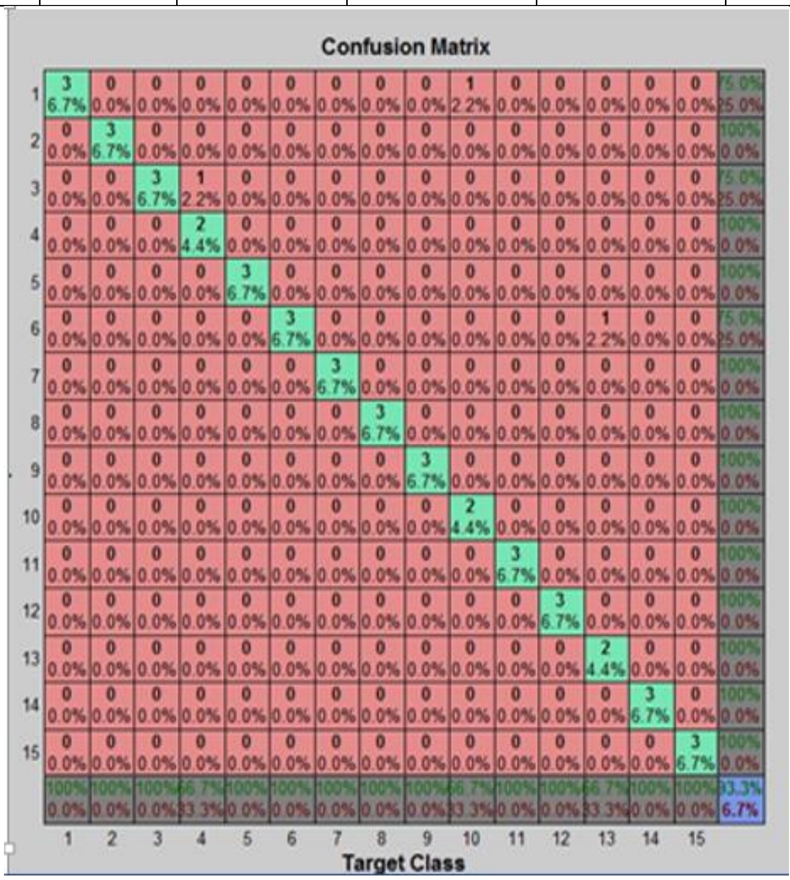

Fig.7. Plot Confusion (against) target Class

6. Conclusion

This paper proposed a neural network based face recognition method. The outputs of the basic PCA are the inputs of the neural network. The experimental results showed the efficiency of the proposed method. In the future we will focus on testing a larger number of variables in the pictures. And we will do further research on the most relevant method to use for the minimization of feature extraction of images that are placed in the database.

7. Acknowledgements

This work was supported by South African National Research Foundation Incentive Grants (No. 81705 and 95687).

8. References

[1] T. Valentine, J.P. Davis. Forensic Facial Identification: Theory and Practice of Identification from Eyewitnesses, Composites and CCTV, Wiley Press, 2015.

[2] N. Lund. Attention and Pattern Recognition, Routledge Publisher, 2001.

[3] L.F. Barrett, M. Lewis, and J.M. Haviland-Jones. Hanbook of emotions, Guilford Publication, 2016.

[4] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. IEEE Transactions on Computers, 1973, 22(1):67 – 92.

[5] R. Hsu, A.K. Jain. Face modeling for recognition, Proceedings of 2001 International Conference on Image Processing, 2001, New York: IEEE, pp. 693-696.

[6] E.J. Candès, X. Li, Y. Ma, J. Wright. Robust Principal Component Analysis? Journal of the ACM , 2011, 58(3), article no.11.

[7] http://www.cl.cam.ac.uk/ Research/DTG/Att archive/pub/att_faces.zip

[8] Http://www.fel.edu.br/face/database 2010

[9] M. Nixon, A.S. Aguado. Feature Extraction & Image Processing for Computer Vision, Academic Press, 2012.

[10] D. Kumar, C.S. Rai, and S. Kumar. Face Recognition using Self- Organizing Map and Principal Component Analysis. Proc. On Neural Networks and Brain, New York: IEEE, pp. 1469-1473.