

# ALGORITHM OF FACE RECOGNITION BY PRINCIPAL COMPONENT ANALYSIS

KHALID A. S. AL-KHATEEB AND JAIZ A. Y. JOHARI

*Electrical and Computer Engineering Department, Faculty of Engineering, International Islamic University. Malaysia (IIUM), Jalan Gombak, 53100, Kuala Lumpur, Malaysia.*

*e-mail: khalid@iiu.edu.my, jaiz@iiu.edu.my*

---

**Abstract:** A face recognition algorithm based on Principal Component Analysis (PCA) has been developed and tested for computer vision applications. A database of about 400 facial images was used to test the algorithm. Each image is represented by a matrix (112 x 92), The data base is divided into subsets, where each subset represents one of 10 different individuals. A 96% rate of successful detection and a 90% rate of successful recognition were obtained. Several factors had to be standardized to provide a constrained environment in order to reduce error. The analysis is based on a set of eigenvectors that defines an Eigen Face (EF). The method proved to be simple and effective. The simplified algorithm and techniques expedited the process without seriously compromising the accuracy.

---

## 1. INTRODUCTION

The face is a natural biometrics, because it is a key component in the way humans remember and recognize each other. The subject of automatic face recognition systems is receiving considerable attention from a variety of interest. The effort has been encouraged by the ever increasing use of high tech-devices in a variety of applications. Face recognition algorithms may in some respects out perform humans, particularly where a large database search is involved. Most of the recently developed methods perform well under constrained conditions. However, the problem of live face recognition remains largely unsolved. Face recognition techniques usually analyze the unique shape, pattern, and positioning of facial features. Using computers for this purpose is a complex technology and largely software based. Some of the problems associated with live real time recognition are complicated by the change of conditions, such as the position of the head and camouflage, which affect the performance of the recognition process considerably. Therefore, to determine a specific face in a group image by computer becomes an extremely challenging problem, whereas the human eye with the built in faculties can detect a face at a glance. To overcome these difficulties and increase the accuracy, intelligent algorithms have to be developed, which can adapt to these changes. Some of the more common techniques are based on neural networks, skin filter, template matching, and eigenvector analysis. The Principal Component Analysis (PCA) technique adopted in this work is based on eigenvectors. The technique was initially developed for the purpose of image compression and reconstruction [5,6]. This technique is suitable for both face and image recognition. The problem of face

detection, i.e. to find the location of every human face contained in an arbitrary still image, represents the first step in a fully automatic face recognition algorithm. It can then be used in an image database indexing or searching by content, for surveillance and human-computer interface systems. It also provides insight, on how to approach other pattern recognition problems. At the same time, it may be one of the challenging problems in the development of pattern recognition algorithms. The present work deals with four aspects of the problem namely, finding a face (input image), track the image in the database, recognizing the face and determining its identity.

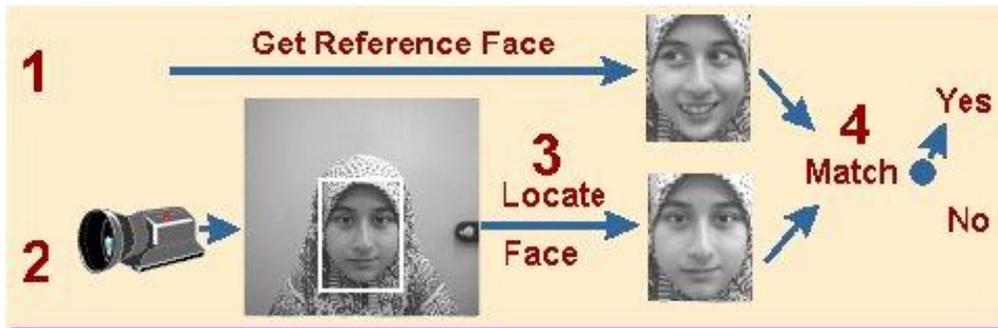


Fig. 1: Process of identifying an image.

## 2. PRINCIPAL COMPONENT ANALYSIS (PCA)

The image is represented by a matrix of smaller images. The smaller images are derived from the full image in order to find the most significant eigenvectors of pixel-wise covariance matrix for a set of training images [1]. The value of a pixel can vary from 0 (black) to 256 (white) depending on its grayscale level. An algorithm capable of recognizing faces based on the Principal Component Analysis or Karhunen-Loeve [6,7] transform is applied. It reduces the amount of processing. Hence the task becomes more manageable. The computer treats a face image as matrix of several hundred by several hundred pixels. If the pixel matrix is turned into a vector, the picture will have a location vector in thousands of dimensional spaces. The small subsection of the multi-dimensional space is called the "face space." The eigenvectors create a new basis to represent the images as coordinates in a much smaller dimensional space. Hence, images of the same person will be located nearer to one another in this space because of their similarity that gives close eigenvalues.

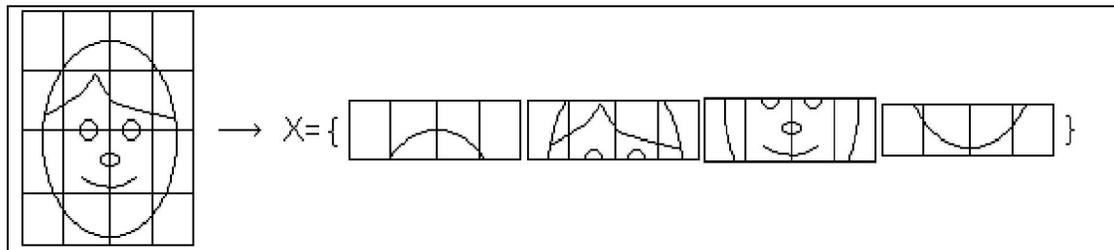


Fig. 2: Formation of the face vector from the face image.

### 3. THE EIGENVALUE PROBLEM

The implementation of the face recognition algorithm for a grayscale image, and the software for the computing processes and the simulation of software interfaces were achieved by programming using MATLAB. The database images had to be standardized in terms of grayscale level, brightness and contrast in order to avoid large error. The theory behind the eigenface approach is derived from the mathematical definition of eigenvectors and the associated eigenvalues;

If  $\mathbf{A}$  is an  $n \times n$  matrix, then a nonzero vector  $\mathbf{x}$  in  $\mathbf{R}^n$  is an eigenvector of  $\mathbf{A}$ , such that  $\mathbf{A}\mathbf{x}$  is a scalar multiplied by  $\mathbf{x}$ ,

$$\mathbf{A} \mathbf{x} = \lambda \mathbf{x}, \quad (1)$$

where,  $\lambda$  is a scalar representing eigenvalues and  $\mathbf{x}$  eigenvector of  $\mathbf{A}$

The eigenvectors of  $\mathbf{A}$ , and the corresponding eigenvalues, are the nonzero vector  $\mathbf{x}$  that satisfy Eq. (1). The eigenvectors corresponding to  $\lambda$  are the nonzero vectors in the solution:

$$(\lambda_i - \mathbf{A}) \mathbf{x}_i = 0$$

i.e. the solution space or the eigenspace of  $\mathbf{A}$  corresponding to  $\lambda$ .

The recognition algorithm begins by collecting a large number of images in a database [6]. The system then creates a set of eigenfaces by combining all the facial images in the database and comparing commonalities and differences between groups of individual facial images. The eigenfaces developed by the system appear as two-dimensional sets of light and dark areas arranged in a particular pattern. Then, the algorithm compares the inputs' eigenface characteristics with those in the database and determines a "degree of fit" score for the target face. If the target face produces a high enough degree of fit score, it is recognized and accepted by the algorithm. In practice, any individual can be identified using a database of 1 to 100 eigenfaces.

A variation of the eigenface approach is called eigenfeatures. The approach combines facial metrics, which involves measuring the distance between specific facial features, such as the eyes, nose, and mouth, with the eigenface approach. The distance between these points on an input live face as measured by the system are then compared to the sets of eigenfeatures stored in the database to determine whether the face is a match.

### 4. ALGORITHM OF THE PCA EIGENVECTORS

The Principle Component Analysis (PCA) method is based on statistical data analysis used in image coding to find a subset of principal directions in a data set of any statistical distribution [5-7]. An image can be represented as a point in an  $(\mathbf{M} \times \mathbf{N})$  dimensional image space, where each dimension is associated with one of the pixels in the image. The values in each dimension are the gray level of the pixels. The notion of direction of variance in a high dimensional space can be extracted from the covariance matrix. The eigenvectors of the covariance matrix define a space in which the dimensions are zero so that the matrix is diagonal. The  $\mathbf{M}$  training images are

represented as points in  $\mathbf{M}$  space. Then, recognizing a face in a new test image would be to find its nearest neighbor in the image space. This approach however is rather slow. The size of image space may be very large and the images are likely to be clustered near one another, since they are all facial images.

For image  $\mathbf{I}_i$  among a collection of  $\mathbf{M}$  images, the average image of all the training images in  $\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_M$  is defined as:

$$\mathbf{A} = \frac{1}{M} \sum_{i=1}^M \mathbf{I}_i \quad (2)$$

The difference vector  $\mathbf{D}_i$ , which shows how every image differs from the mean, can be calculated.

$$\mathbf{D}_i = \mathbf{I}_i - \mathbf{A} \quad (3)$$

The algorithm of covariance matrix  $\mathbf{C}$ , of the data is thus defined as:

$$\mathbf{C} = \frac{1}{M} \sum_{n=1}^M \mathbf{D}_n \cdot \mathbf{D}_n^T \quad (4)$$

$\mathbf{C}$  is very large, so this method is computationally intensive. However, there are relatively fast methods for finding the  $\mathbf{k}$  largest eigenvector, which is all that is needed. The  $\mathbf{M}'$  images;

$\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_{M'}$ , are called eigenfaces or eigenvectors [4,6]. These images define a basis set, so that each face image will be defined in terms of how similar it is to each of these basis images. For  $\mathbf{k} = 1, \dots, \mathbf{M}'$ , a real-valued weight,  $\mathbf{w}_k$  can be computed, indicating the similarity between the input image  $\mathbf{I}$ , and the  $k$ th eigenvector,  $\mathbf{E}_k$ . Solving this eigenvector algorithm problem, any image  $\mathbf{I}$  can be projected into eigenspace by  $\mathbf{w}_k$ , that is,

$$\mathbf{w}_k = \mathbf{E}_k^T (\mathbf{I} - \mathbf{A}) \quad (5)$$

The algorithm of eigenvalues  $\lambda_k$ , can be calculated as:

$$\lambda_k = \frac{1}{M} \sum_{n=1}^M (\mathbf{E}_k^T \mathbf{D}_n)^2 \quad (6)$$

and defining the algorithm of eigenvector as:

$$\mathbf{E}_l^T \cdot \mathbf{E}_k = \delta = \begin{cases} 1, & \text{if } l = k \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Thus, any image  $\mathbf{I}$  can be expanded as a linear combination of  $\mathbf{E}_k$  and  $\mathbf{w}_k$ , where  $\mathbf{w}_k$  describes the contribution of each eigenvector.

$$\mathbf{I} = \mathbf{A} + \frac{1}{M} \sum_{j=1}^M \mathbf{w}_k \cdot \mathbf{E}_k \quad (8)$$

The accuracy of the eigenspace method as characterized by the performance of the eigenspace algorithm resulted in about 96% correct recognition. If a training set contains multiple images for each person, then the average point in eigenspace can be computed and the computed points represent each image of that person. The algorithm requires that all images of a 3D object such as a person's head or face from many different position, orientations and expression. The points in eigenspace corresponding to a different 3D view can then be combined by fitting a hyper surface to all the points, and storing this hyper surface in an eigenspace as a description of that person. Then, classify a test image (input image) as a person corresponding to the hyper surface.

The algorithm of reading and matching the eigenfaces examines a large number of pictures of faces are stored in a database. Any facial image is built with different intensities of light. The lighting condition however must be controlled to be the same for all images. To identify a face, the algorithm compares its eigenface characteristics, which are encoded into a template, with those in the database, so as to select the face whose templates match the target most closely. To verify someone for example, a claim is first made for the person. This claim is in the form of a user ID that can be represented as a 2D bar code, card, token or smart card. The user ID claim is used to fetch the face template representing the true identity of the person (reference template), which is stored in a database. When the true face matches the known reference image in the database against the input image it produces a match score. For a match score greater than a preset threshold, the two faces are deemed to be for the same person. Otherwise, they belong to another person. In the identification of an input image against a database and then all the match scores are sorted from the best score down to a percentage error is less than 4%. If the top score is above a matching threshold, then it is the best candidate to be identified the unknown person (input image).

## 5. CREATING AND STANDARDIZING THE IMAGE FILES

In this algorithm, the input and database images represented as  $\mathbf{M} \times \mathbf{N}$  matrix (112 x 92) pixels. The image is defined as a grayscale image in which case the data range is from 0 to 255. The image is formatted in TIFF or GIFF file format. Each image would have a pixel intensity arrangement of 2,637,824 possibilities. This gives a better accuracy, and the face recognition algorithm will be more reliable. The procedure begins by selecting only the face part from the original image. Then, the brightness and contrast are standardized. The color photo is turned into a grayscale image. The size is changed to (112 x 92) pixels, and saved in TIFF file format. Some of the processing steps are illustrated in Fig. 3 and Fig. 4. The image information is standardized in terms of size and color intensity. The factors observed during the creation of the files are:

- 1) The distance between face and lens, and the light intensity are kept constant. This is especially important in creating images files for the same person.
- 2) Each image is chosen to correspond to a different facial expression. This enhances the reliability of the system.
- 3) When converting the color image into a grayscale, the average intensity for each image in database must be specified.

The process of standardizing the brightness and the contrast is important because the conversion of the original color image into grayscale may introduce error, since the grayscale contains only 256 levels from black to white.

## 6. RECOGNITION ALGORITHM

The face recognition is the final stage of the process and is used for determining both the closest match for the identity of the person and the probability of that match. The algorithm compares the input image with every image in the database. The image in the database would have undergone similar process as the input images.

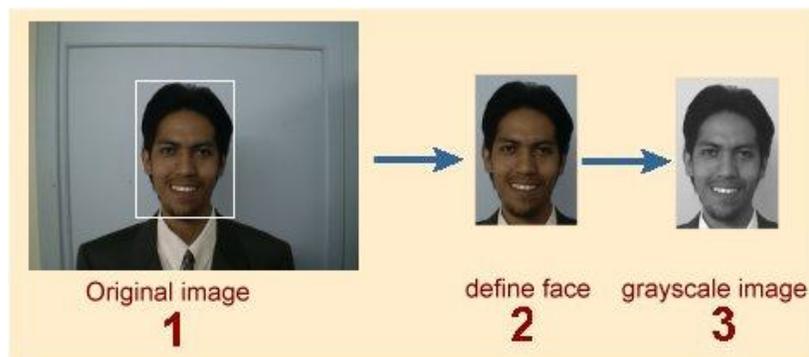


Fig. 3: Steps of creating database and input image.

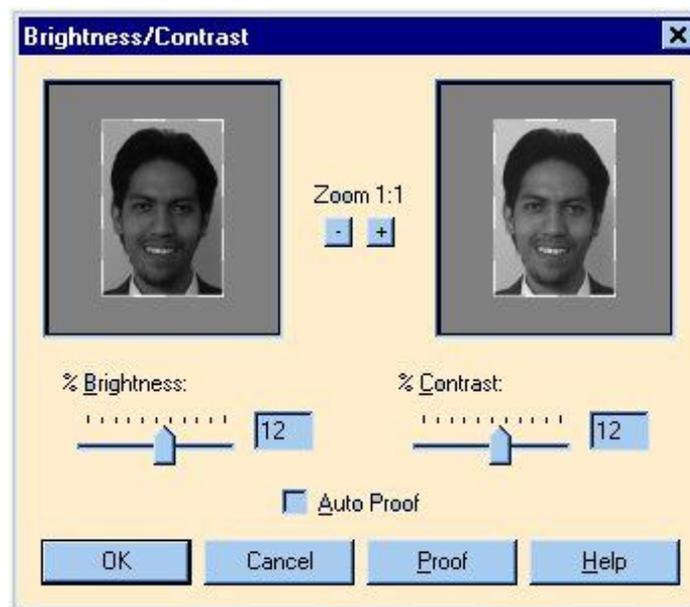


Fig. 4: Standardizing the brightness and contrast.

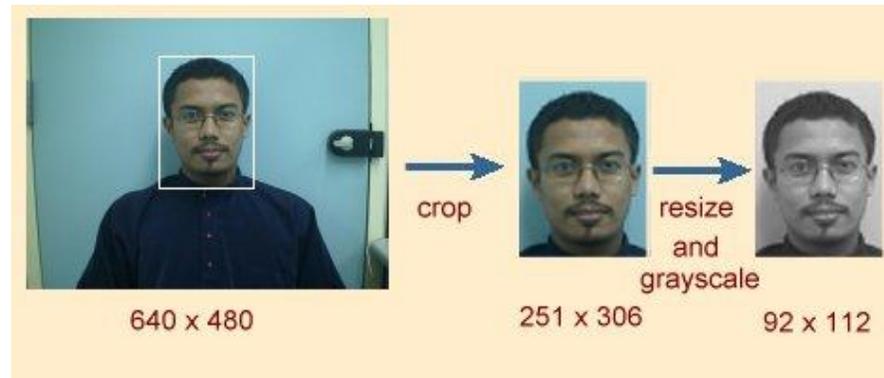


Fig. 5: Process of crop and resize

A singular eigenvalue will then be used to represent the input image. The eigenvalues will then be compared with the eigenvalues of the database images. A match is acquired only when the margin of error is less than 4 %.

The algorithm of recognition is considered a failure if the error is greater than 4 %. Furthermore, the number of eigenvalues can be as many as the total number of images in the database, which may cause a problem. This problem was solved by creating a new version of the set of eigenfaces represented by the eigenvalues. It is achieved by removing the images which conform least to the set. The result is reflected in enhancement of the recognition accuracy of the algorithm. Of course, higher accuracy could still be expected with more accurate algorithm calculations rather than approximations. The improvement, however, will be at the cost of extra processing time.

## 7. TESTING AND ACCURACY

A number of testing procedures were adopted in order to find the optimum number of images for a person that gives highest recognition probability. Several database sets were created, each comprising a different number of persons. The recognition percentage was standardized in each case, and the recognition is considered successful only when the match is above 96%.

An example of the performed tests is shown in Table 1. Note that the images for each person were selected at random to test the algorithm accuracy.

The highest accuracy obtained during the test was 90 %, when each individual was represented by five pictures. The lowest accuracy was 10 % when each individual was represented by 9 images. The problem when there are too many images representing the same person is that the eigenvalues will be closely spaced. Hence, the eigenvalues images belonging to different persons may overlap. On the other hand, when the number is too small, the information will be insufficient to give acceptable recognition accuracy. In this case, five images seem a reasonable number, as shown in Fig. 6. While the information is sufficient to represent an individual, the values are not too closely spaced as to cause confusion. A similar test was performed on images with different type of extension PCX and JPEG. For this test, two extra databases were built the content of each were images from different extensions. The PCX and JPEG images

were based on the original TIFF images. The result of the test (Fig. 7) shows the similarity between TIFF and PCX in contrast with JPEG.

Table 1: Test results of random image.

	Number of Images for each Person								
Person	1	2	3	4	5	6	7	8	9
S1	No	No	Yes	Yes	Yes	No	Yes	Yes	Yes
S2	Yes	Yes	Yes	No	Yes	Yes	No	No	No
S3	No	No	Yes	Yes	Yes	Yes	No	No	No
S4	No	No	Yes	Yes	Yes	Yes	No	No	No
S5	Yes	Yes	No	Yes	Yes	No	No	Yes	No
S6	No	No	No	No	Yes	No	Yes	No	No
S7	Yes	Yes	Yes	No	Yes	No	No	No	No
S8	No	Yes	No						
S9	Yes	No	No	No	No	Yes	No	No	No
S10	No	No	Yes	Yes	Yes	No	Yes	Yes	No
	40 %	40 %	70 %	60 %	90 %	50 %	40 %	40 %	10 %

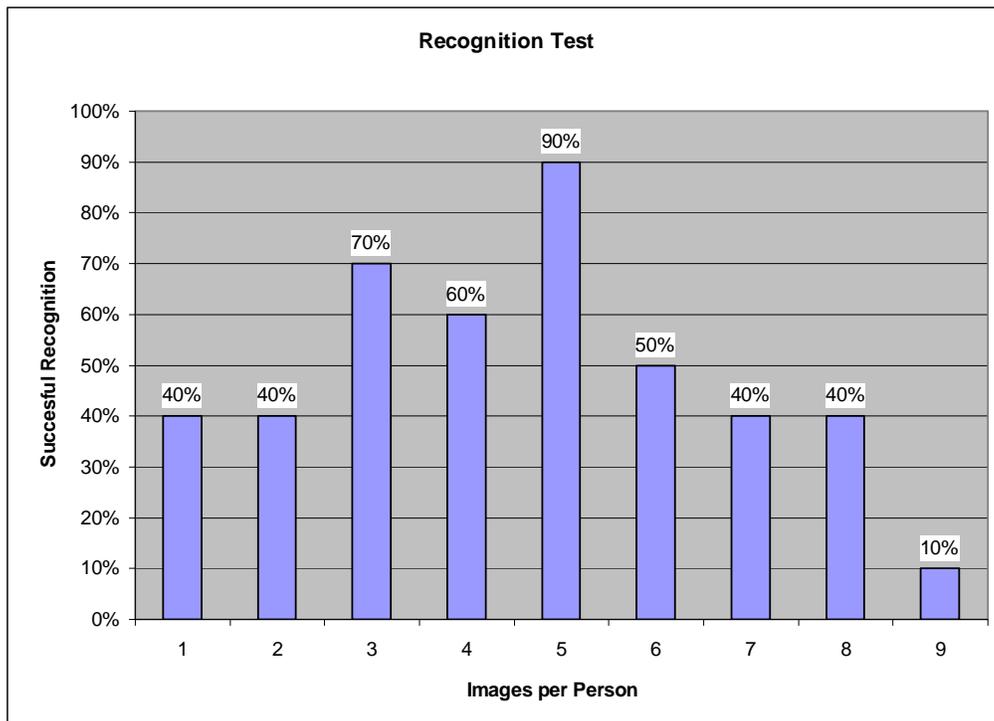


Fig. 6: Result of recognition versus no. of images representing one face.

The PCX and TIFF images have similar accuracy. The images do not encounter much loss of information during the exchange. As observed from Fig. 7 JPEG suffers changes and encounters some losses during transformation. The JPEG is intended for compressing images that will be looked at by humans, where small errors do not cause a problem. These small errors (introduced by JPEG) may be a problem for images analyzed by machine, even if they are invisible to the eye. Below is an example between the original images (TIFF) which has undergone the JPEG transformation. Slight changes can be recognized between the two images. The TIFF image seems to have more details than the JPEG. When these two images are analyzed in terms of pixel values, some differences are noticed. The grayscale values go from 29 higher to -34 lower. This proves that the JPEG image undergoes a change during transformation.

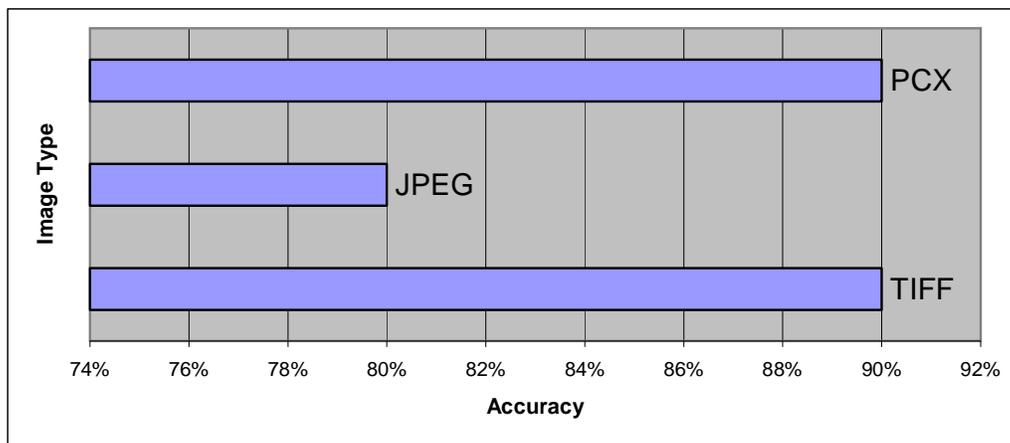


Fig. 7: Accuracy with the different extension.

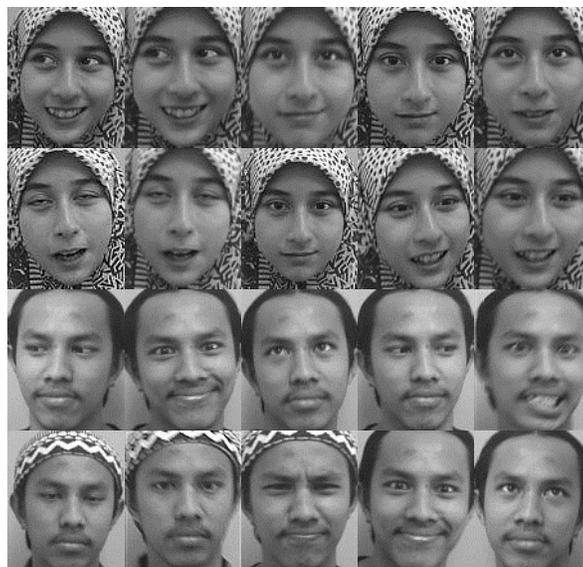


Fig. 8: Part of the face images used to create eigenspace.

The accuracy of the recognition system relies greatly upon how the eigenface algorithm is implemented. There are shortcuts and approximations that can save time and resources. However, this would be at the expense of accuracy. The eigenvalues and the covariance matrix are among the values that can be calculated in several ways. All the recognition algorithms are built on how these values are computed. The covariance matrix for example can be solved using the full algorithm computation method, which requires very large memory space. Sometimes approximate values are sufficient, and these require less memory space and are much faster. The number of images in the database also plays an important role in determining the accuracy of the recognition, as discussed earlier. Other factors affecting accuracy depend on differences in the design and structure of the system, since different programmers have their own ways of presenting the system.

The algorithm of recognition by PCA is reasonably fast and has several advantages such as, the raw intensity data are used directly for learning and recognition without a significant low-level or mid-level processing. The geometry and the reflectance of a face are not a requisite. The low-dimensional subspace representation suffices for data compression. The recognition process is rather simple and may be more efficient by comparison with other methods. On the other hand, the method has its limitation and drawbacks. Theoretically, the constrained conditions are reasonably simple. Experimentally, however, they may present serious limitations resulting in a number of disadvantages. For example, the method is sensitive to scale. Therefore, a low-level preprocessing may still be necessary. The eigenface representation is faithful to the original images, in the least-square sense, but the recognition rate decreases under varying pose and illumination conditions. It has shown to be robust when dealing with facial expression and some disguise effects e.g glasses, but the tests were made with frontal view only. Of course, the problem will be much more difficult for extreme changes pose, expression and disguise. The eigenface recognition method being "appearance-based" shares some drawbacks with other methods. First, it is difficult to update the database, because of the time-consuming learning for training images. Second, the recognition is efficient only when the number of face classes is larger than the dimensions of the face space, otherwise the projection of an unknown image requires pixel by pixel multiplication of the input image by all eigenfaces, which is equivalent to or worse than template-matching.

The current face recognition algorithm may work under constrained conditions. However, they nearly all fail under vastly varying conditions. In the future, recognition algorithms will need to recognize people in real-time and in much less constrained situations. Improving the current algorithm may prove to be advantageous in Real Time Recognition. It would first involve finding a face within an image, which in any real time situation a face can be seen from various angles at different distances from the camera. The recognition algorithm must be able to identify which part of the image captured is a face for later processing, and the normalization of illumination and other factors.

## 8. CONCLUSIONS

The algorithm of face recognition by Principal Component Analysis (PCA) is a rapidly developing area with many possible applications from crowd surveillance to improved

human-computer interaction. The goal of this work is to develop a recognition algorithm and investigate the issues involved in getting the algorithm to work. Face recognition using eigenface algorithm by PCA was evaluated with a number of images and shown to have a maximum recognition rate of 90%. The relationship between the recognition accuracy and type of image was investigated. It is found that the compressed image resulted in a slightly decreased performance. Multiple techniques are available in obtaining values such as the covariance matrix and the eigenvalues. Future work will continue be introduced to improve the existing algorithm. There are other aspects that can also improve the current algorithm such as real time recognition and the smart card implementation. Face recognition by eigenfaces algorithm offers significant potential for commercial applications which makes it well suited for future development.

## REFERENCES

- [1] Khalid A. S. Al-Khateeb and Jaiz A. Y. Johari, "Implementation of Eigen Vectors in Computer Face Recognition", 8<sup>th</sup> Babylon Academic Conference, Babylon, Iraq, 23<sup>rd</sup> April 2002.
- [2] H. Anton and C. Roses, "Elementary Linear Algebra Application Version", John Wiley and Sons Inc., U.S.A, 1994.
- [3] J.C. Russ. "The Image Processing Handbook, MATLAB", Second Edition, CRC Press, Boca Raton, 1997.
- [4] M. Turk and A. Pentland "Eigenfaces for Recognition", Journal of Cognitive Neuroscience Vol.3, Nov 1991 pp 71-86
- [5] R. Cappelli, D. Maio and D. Maltoni "Multispace KL for Pattern Representation and Classification", IEEE Transactions and pattern analysis and machine intelligence Vol.23, 2001 pp 977-996
- [6] M. Gross and H. Luttermann, "Combining Principle Analysis and Neural Networks for the Recognition of Human Faces", Proceeding of Pacific Graphics, Vol. 1, 176-192, 1993.
- [7] Z. M. Hafeed and M.D. Levine "Face Recognition Using the Discrete Cosine Transform" International Journal of Computer Vision 43(3), pp 167-188, 2001.
- [8] B.J. Oommen, K.S. Loke "On the Pattern Recognition of Noisy Subsequence Trees" IEEE Transactions and pattern analysis and machine intelligence, Vol.23, pp 929-946, 2001.

## BIOGRAPHY

**Prof. Khalid al-Khateeb** studied in the United Kingdom and graduated at the Royal College with an Honors B.Sc. degree in Electronics in 1966, the Master's degree in 1971 at Salford University and the PhD in 1975 at Manchester University.

His research interests include electronics, communications, computer applications and engineering education. His academic life stretches over decades, during which he worked at various universities in the UK, USA, Iraq, Algeria, Jordan and Malaysia.