Face Recognition Using Principal Component Analysis (PCA) and Shape Descriptors

Firstname A. Lastname

Name of Institution/Department, City, Country

Email: first.author@hostname1.org

***Abstract*— In recent years, face recognition has received substantial attention from both research communities and the market, but it is still a very challenging area. Face recognition is the process of recognising people by their facial image. It is an important application in security systems that involves authentication or confirmation of an identity. Face recognition consists of three parts: firstly, face detection: this stage is considered to be done in our research; secondly, feature extraction and, finally, classification. We concentrate on feature extraction methods as the performance of a face recognition algorithm depends on which feature extraction method is used. Facial feature extraction approaches can be categorised into a) holistic approaches such as Principal Component analysis (PCA) and b) geometric feature-based approaches such as shape descriptors. This paper describes face recognition using PCA algorithm and shape descriptor methods as feature extraction methods and the Euclidean distance classifier. It also describes how to implement face recognition algorithms using the PCA method and Hu's seven moment invariants, which is an example of the shape descriptors method. A comparison between the two approaches is also considered.**

***Index Terms*—Face recognition, PCA, shape descriptor, Hu's seven moment invariants.**

I. Introduction

Face recognition has been an active research area for more than 10 years. "The face is the primary focus of attention in the society, playing a major role in conveying identity and emotion. Human face recognition plays a significant role in security application for access control and robotics [1].

Direct identification in face recognition is inefficient for many reasons. First of all, an image consists of a huge amount of data. Therefore, it utilises large memory space as well as being time consuming. There is much information in an image, which is related to the identification of an image; this information should be extracted and selected. This identified data is used in face recognition. The aim of feature extraction is to reduce the image dimensions and reduce computation time and identification error by transforming high dimensionality of the image into lower dimensionality [2].

Dimensionality reduction is reducing the dimensions of an image by extracting meaningful (statistical) information and representing it in low dimensions, which in turn facilitates the recognition stage [3]. Several methods are used in feature extraction to identify which feature of a face image is related to recognition, for example, Principal Component Analysis (PCA) [2].

Techniques of face recognition fall under two categories:

- Holistic approaches
- Feature-based approaches

Holistic approaches take into account the whole face image when extracting face features. PCA is applied in order to have low dimensional representation of face images. In feature-based approaches, individual feature (local feature) points such as eyes, nose and mouth are detected and the distance between them is calculated to build feature vectors [4].

In this paper, we introduce the concept of facial feature extraction using PCA algorithm. The remaining part of the paper is organized as follows: Section III outlines the procedure of implementing face recognition using PCA algorithm. Section IV describes the concept of extracting face features using Hu's seven moment invariants, which is an example of the shape descriptors method. Section V presents the details of the implementation of face recognition using Hu's seven moment invariants. Finally, section VI presents the study's conclusion and compares the two research methodologies, the PCA and Hu's seven moment invariant.

II. FACIAL FEATURE EXTRACTION USING PCA

Face recognition has been in the research area since the 1960s. In 1960, Bledsoe and other researchers worked in this area [See 5]. In 1964, how a computer can recognise faces was introduced by Bledsoe. Research in 1980s and 1990s formed the basis of the current face recognition techniques. PCA was introduced by Karl Pearson in 1901. The idea of using PCA to represent face images was introduced by Kirby and Sirovich [6].

In 1991, PCA became a popular method used by Turk and Pentlandin for face recognition. PCA is used to extract significant features of face images in feature space. This is called "eigenfaces" and training images in the database are projected onto a lower dimensional subspace. Recognition is done by comparing the distance between the new image and image in the subspace (eigenspace) [7].

PCA is a mathematical algorithm so it is easy to describe and use. Moreover, it is one of the most efficient techniques in terms of reducing dimensionality of a dataset (data compression). Therefore, this leads to facilitating image classification [8].

A. Mathematical representation of PCA

2D training images are represented in 1D vector T by concatenating the rows of each image into a single vector. It is of size L (N x M), where N is the number of rows and M is the number of columns. Suppose there are M training images. Therefore, the vectors of training images are **(1)**

The average vector of T represents mean face

(2)

Subtracting the average vector Y from image vectors where. **(3)**

Form the matrix. A matrix contains only the different features for face images. This implies that it distinguishes the variance of face images. Size of

Calculate the covariance matrix according to the next equation:

The size of C matrix is

Practically, the data is large and it is difficult to compute eigenvectors of this matrix C. To deal with this problem, eigenvectors of matrix are computed. L matrix has smaller data as its size = M x M. So, eigenvectors of C are computed from eigenvectors of L.

We suppose and are eigenvectors and eigenvalues of L respectively. According to the following equation,

Highest eigenvalues are used to select the eigenvectors that reflect the greatest variance in the face image.

If we multiply two sides by A, we get the following:

According to the above equation, are eigenvectors and eigenvalues of matrix C respectively? Matrix C can have eigenvectors. Matrix L can have M eigenvectors. These vectors form Eigen faces

Face Space

Eigenfaces are formed face space [11]. The face image is then projected in eigenface space by using the following formula:

Where **(8)**[9][10]

The weighted matrix that represents the projected face image is = $[w1, w2 \ldots \ldots wm']^T$ **(9).** Ω contains the eigenfaces that represent the base for face images [10].

Recognition

To perform recognition of a new face, this image is transformed into eigenfaces. After that, the Euclidean distance is used to compute the distance between face classes and the new image. In order to find which face class is the best match for the new face image, the smallest distance is considered.

Where Ω k is the vector represent k face class [9].

III. IMPLEMENTATION OF FACE RECOGNITION USING PCA ALGORITHM

This section introduces a process of implementing face recognition using a PCA algorithm.

A. Problem Definition

The problem of face recognition can be defined as searching a face database for a given face to find an equivalent face, which identifies the person [12].

B. Database Design

In face recognition, a large number of available databases exist for researchers such as ORL face database [13]. In this research, my database will be used for the implementation of the face recognition system using the PCA algorithm.

My database consists of 15 images of 3 individuals (one male and two females). This means that the database contains images grouped in three sets. There are five different images for each person. These images were captured in a variety of poses with different lighting conditions, which are the most important factors in a database. All images were taken against a white background.

The database that is used to implement face recognition is grouped into separate training and testing databases. The training database contains 15 images. These images are used to train the algorithm in the training phase. See Figure 1. The second set is the testing database, which is used in the recognition (testing) phase. There are three images in the testing database, one for each person. These images are chosen from the training database. See Figure 2.

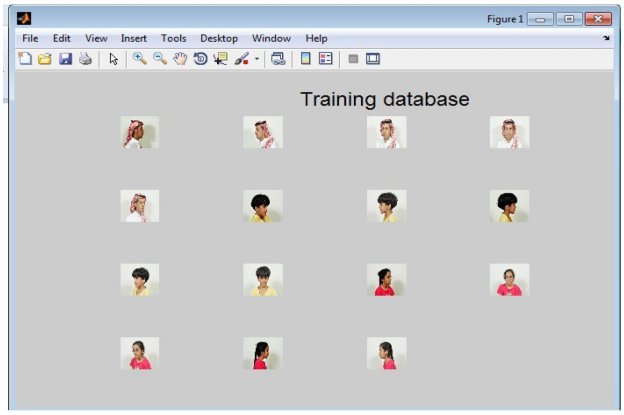

Figure 1. Training database

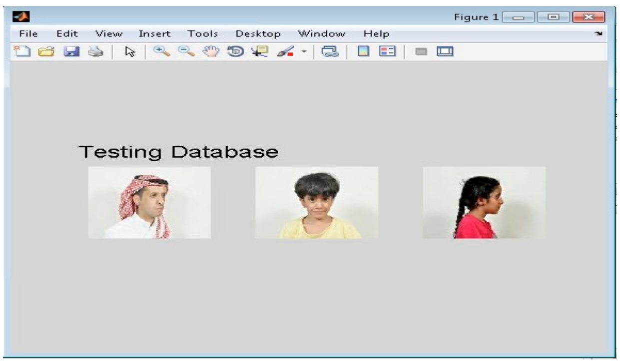

Figure 2. Testing database

C. Working Procedure

There are two phases that form the PCA method. These are the training (learning) and testing (classification) phases [9]. These stages work on the face database.

Training phase

In the training (learning) stage, a new coordinate system is identified based on large eigenvectors by applying the PCA method on the training set database so that face space is constructed [9].

In order to create the face space, several steps should be performed (See Figure 3). First of all, the train database path is selected. Then the ***Function T = Create Database(TrainDatabasePath)*** is invoked. This function gets TrainDatabasePath, which contains the path of the training database as an input argument. It converts 2D training images in the face database into 1D column vectors. In order to construct a 2D matrix that has all 1D image vectors, 1D column vectors are set in a row. Suppose the number of training images in the face database is M and all have the dimensionality, then the size of the 2D matrix T is [14]. There are 15 training input images, five per person. The size of each input image is. Therefore, the 2D matrix, which represents training images, contains 36,000 rows and 15 columns. Secondly, the PCA algorithm is applied to the training image vectors. As a result, eigenfaces that reflect the greatest variance in the face images are identified. ***Function [m, A, Eigenfaces] = EigenfaceCore(T)*** is invoked to calculate these eigenfaces. This function takes T, which is a 2D matrix that represents 1D image vectors as an input argument. The output arguments are m, A and eigenfaces, where m is the mean of the face training set, which contains the common face features. The size of the mean face is. Once m has been calculated, the mean vector m is subtracted from all 36,000 dimensions. Therefore, normalised training set A is calculated. This in turn removes common features of the training set. The size of centered images matrix A is. As mentioned in section II, eigenvectors are calculated from, which has smaller dimensions than. The eigenvectors of L are then sorted and the highest eigenvectors, which exceed the threshold value, which is equal to 1 in this code, are kept; other eigenvectors are eliminated. Moreover, these selected eigenvectors of matrix L are used to calculate the eigenfaces (eigenvectors) of covariance matrix C so the face space is established. The size of eigenfaces is. [14][15].

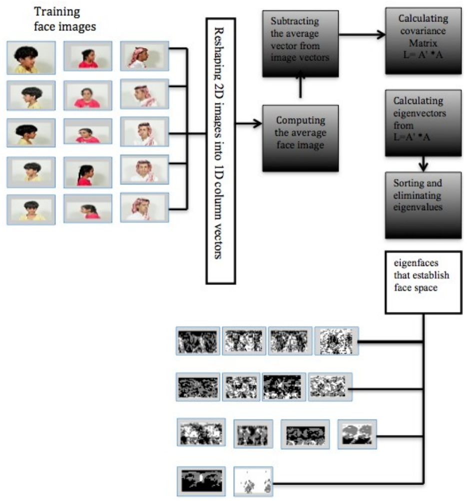

Figure 3. Block diagram of training phase

Testing phase

In testing (classification) phase, normalised training set A images and the test image are projected into the same coordinate system (face space), then the Euclidean distance is used to compute the distance between these projected training images and the testing image [14].

First of all, the test database path is selected. After that, a user enters the name of the test image, which is a number between 1 and 3. Secondly, ***Function OutputName = Recognition(TestImage, m, A, Eigenfaces)*** is invoked. It takes four input arguments: TestImage, which contains the path of the test image; m, representing the mean of the face training images (output from EigenfaceCore()); A is the normalised training set (output from Eigenface Core()); and Eigenfaces, which are Eigenfaces of covariance matrix C of the training set (output from EigenfaceCore()). It returns the name of the equivalent image to the test image in the training database, which is assigned to the OutputName argument [14].

In the recognition function, several steps are performed. Firstly, all normalised image vectors of training database are projected into the eigenface space. This is done by multiplying Eigenfaces' by these normalised images. As a result, the feature vector of each face is identified. Secondly, the mean vector m is subtracted from the test image vector. The test image is then projected to the same eigenface space () and the feature vector of the test image is identified. Finally, classification is done using a Euclidean distances classifier. Euclidean distances are calculated between the feature vector of a test image and the feature vector of each face in the training database. Finding the minimum distance between the test image and its match image in the training database is considered to be the best match. This step is considered to be the core of the algorithm [14] (See Figure 4).

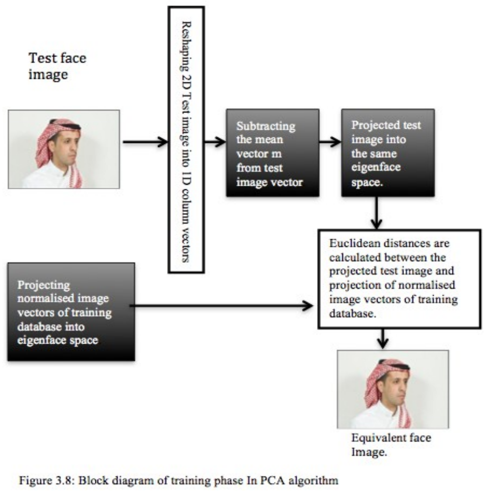

Figure 4. Block diagram of training phase in PCA algorithm

Result and discussion

The Graphical User Interface of our face recognition presents the three functions (CreateDatabase, EigenfaceCore and Recognition) described above, in addition to an Exit button. The user simply clicks on the CreateDatabase button to construct a 2D matrix of 1D vectors of the training database. By clicking on Calculate Eigenfaces, face space of the training set is established. When the user clicks on the Recognition button, Test and Equivalent images are displayed, followed by displaying the name and age of the person whose image is recognised in the training database. The Exit button is used to terminate the face recognition system.

The result of our face recognition system is shown in Figure 5. To find the equivalent face image, the user is required to input the name of the testing image (a number between 1 and 3) after the selection of the test database path. The algorithm determines the equivalent image of the testing image based on Euclidean distances. Image No 5 in the training database has a shorter Euclidean distance to the test image compared with other training images and is selected to be the equivalent image. Once the user inputs the name of the testing image, the test and equivalent image are displayed in a specified place in the GUI, along with the name and age.

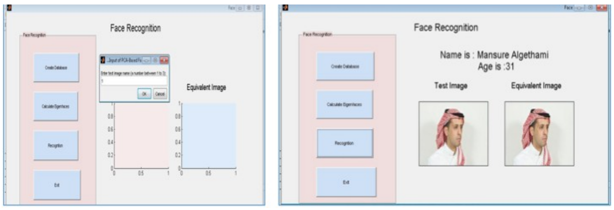

Figure 5. Result of face recognition system

Performance analysis

In order to evaluate the PCA algorithm on our face database based on the Euclidean distances classifier, several experiments were performed. The recognition rate of the algorithm is performed using the following formula:

In the first experiment, we determined the number of training images in the face database that give the best recognition rate. In the testing phase, the images of three people are used. These images are presented in the training database but with different poses and lighting conditions, and the number of images is increased from 1 to 4. In the training phase, we start this experiment with one image per person. The number of images in the training database is increased by 1 until the best recognition rate is obtained [10].

Observation: When the number of images per person is increased, the accuracy of the algorithm is also increased. Adding the fifth training image, which is equivalent to the test image, contributes to getting high recognition.

In the second experiment, performance is calculated in terms of processing time (elapsed time). It is calculated during both the training and testing phase of the PCA algorithm. Elapsed time is evaluated while increasing the number of training images by 1.

The command was used to measure the elapsed program time [16].

Observation: Increasing the number of training images to have the best accuracy will increase the computation time in both the training and testing phase. It is also seen that elapsed time in the testing phase is less than elapsed time in the training phase.

No of the images in training database (per person)	Elapsed time in training phase (Seconds)	Elapsed time for each testing image (No. 1,2 and 3)	Elapsed time in testing phase (Seconds) Elapsed time= $\frac{\text{Elapsed time for all testing images}}{\text{the number of testing images}}$
1	17.0	No.1 = 11.4	36.4 / 3 =12.1
		No.2 =12.3	
		No.3 =12.7	
2	19.3	No.1 =11.4	38.6 /3 = 12.9
		No.2 =13.4	
		No.3 =13.8	
3	21.4	No.1 =11.6	39.4 / 3 = 13.1
		No.2 = 13.7	
		No.3 =14.1	
4	22.6	No.1= 12.2	41.2 / 3 = 13.7
		No.2 =14.4	
		No.3 = 14.6	
5	25.1	No.1= 12.4	42.3 / 3 = 14.1
		No.2= 14. 7	
		No.3= 15.2	

In the third experiment, we select eigenvectors that reflect the greatest variance between face images and eliminate those which have less information about image variation because the images in the database are captured in different poses and lighting conditions and none of these features are used to classify an image. Moreover, each eigenvector holds specific characteristics such as lighting [17].

In order to choose eigenvectors that correspond to the highest eigenvalues, firstly, non-zero eigenvectors are selected. These eigenvectors are then sorted to match the corresponding eigenvalues from largest to smallest. We start this experiment by selecting the first five ordered eigenvectors; this number is then increased until the best recognition rate is achieved.

Observation: Increasing the number of eigenvectors will increase the accuracy of the algorithm. It can be seen that using the 14 highest eigenvectors enhanced the algorithm to achieve a high recognition rate.

However, performance of the algorithm depends on many factors that influence the recognition of face images. These factors are, firstly, the database which is used in the experiment, as the quality of the image contributes to enhancing or deteriorating the algorithm performance. As a result, the better the image quality, the better the feature extraction and the more accuracy we get [18]. Moreover, facial expression, pose angle as well as lighting environment have an effect. As a result, variable poses and lighting reduce the algorithm performance.

IV. Facial Feature Extraction Using Shape Descriptors

Extracted features using holistic and feature-based approaches are different. Extracted features using holistic approaches are used to explain the intensity value of pixels in the original face images. Extracted features using feature-based approaches capture the local face images such as eyes, nose, mouth, chin and head outline [32]. The benefit of holistic approaches over feature-based approaches is that they extract the features which discriminate between face images so good performance is achieved. However, variations in illumination, scale and pose reduce the performance of a holistic face recognition algorithm. The advantage of feature-based approaches is that feature points are identified automatically so they are more robust to pose, lighting and facial expression variation than holistic methods [19].

Since feature-based methods use local facial features, which are used in the shape descriptor methods, which will be described later in this chapter, it is possible to say that shape descriptor methods are examples of feature-based methods.

Using the PCA feature extraction approach in face recognition is not robust to pose variation as well as lighting variation. Moreover, Blackwell et al., as cited by El-Bakry [20], state that PCA, which uses the whole face image, does not substitute the need for using a large data set in the learning stage. Processing the whole face image is the reason for the above limitations in a face recognition algorithm using the PCA method. Using the constituent part of the face image may solve these problems [20]. Features which are used to describe an object (face) can be extracted from boundary, texture, surface or other features. Such features that are used to classify objects are represented using quantitative measurements, which are called **descriptors**. The computational methods which are used to measure shape are called **shape descriptors** [21, 22]. Many approaches of shape representation (descriptors) have been found in the literature such as Fourier Descriptors and moment invariants. In this section, "using shape descriptors methods, which play an important role in feature extraction" in face recognition, will be introduced [23, 24].

A. Shape Descriptors

The shape of an object presents an important aspect in object recognition [25]. The shape of an object of interest (face) is expressed in a mathematical way which is known as a shape descriptor. Shape descriptors are used to assist in the identification of facial characteristics for matching purposes and are used by image processing applications within the field of computer vision. Shape descriptors can be considered as mathematical methods which are applied to an image. As a result, a particular feature of that image is obtained. These features are numerical values, which are defined based on the way in which shape descriptors extract shape features. Once shape features are obtained, a shape vector is used in the recognition phase to classify an input image [21]. Shape description is a cornerstone technology for Content-Based Image Retrieval (CBIR), which involves the search for images within large databases [26].

Shape descriptor methods can fall into boundary-based shape descriptors and region-based approach. Boundary-based shape descriptors exploit the boundary information to represent the shape. Feature vectors, which represent boundary information, can be extracted from the integral boundary [25]. To measure similarity between feature vectors a metric distance can be used [27]. There is another way to extract boundary information, by segmenting a shape's boundary into segments (primitives) by using specific criterion. Many boundary-based shape descriptor methods have been found in the literature, such as Fourier Descriptors and Wavelet Descriptors [25].

In the region-based approach, shape is represented by utilising all pixels within the shape's region, unlike the boundary-based approaches, which only exploit the boundary information. It is a widely used approach as it is mathematically sufficient for describing shape. Furthermore, the benefit of this technique over the boundary-based approach is that unconnected shapes can be represented by using pixels within the shape. However, when there is a need for boundary information in a recognition application, the region-based method is not suitable, as it does not determine the boundary information. Moment invariants are an example of region-based methods [25].

The main advantages of moment features is their invariance to translation and scaling, as well as rotation. Moreover, it is a useful technique that has been used in pattern recognition due to their sensitivity to the pattern features. In 1961, algebraic moment invariants were first defined by Hu. He used these moments in the recognition of alphabetical characters [28, 29]. Since the inception of Hu's moments, they have been used in many areas of research. In 1977, Dudani et al. used invariant moments in recognition of six types of aircraft and they achieved a great performance. In 1990, El-Khaly and Sid-Ahmed used moment invariant descriptors for the purpose of recognition of Arabic characters. In 1994, Ezer et al. compared the use of Fourier descriptors and moment invariants in recognition of planar shape. Also in 1994, Flusser and Suk derived affine moment invariants and used them in handwritten character recognition. In 2002, Haddadnia et al. implemented the moment invariant which is known as pseudo-Zernike in face recognition algorithms. In 2008, Nabatchian et al. studied different moment in face recognition.

B. Using Moment Invariants for Face Recognition

Moment invariants are useful in shape recognition. Different moment invariants have been applied to face recognition algorithms. First of all, the first order of **Regular moment invariants** (RMI) is obtained by calculating the centre of an image. Moreover, algebraic invariants theory is used to derive **Hu moment invariants** (HMI). Furthermore, Zernike orthogonal polynomials represent the base of **Zernike moment invariants** (ZMI) and **pseudo-Zernike moment invariants** (PZMI) [28].

Hu moment invariants

Based on the theory of algebraic invariants, "Hu derived relative and absolute combinations of moment values." [30, p.12]. These moments are invariant to scaling, translation and rotation. First of all, normalisation of size is the result of size invariant moments, which are derived from algebraic invariants. Moreover, the central moment is invariant to translation. Furthermore, Hu proposed two methods to obtain rotation invariance, as it is difficult to achieve. The first method is based on the principal axis of distribution, which is used to compute moments. Therefore, moment which is invariant to the rotation of the distribution will be achieved. Absolute moment invariants are the second method for rotation invariance, which is widely used in moment research [30]. Hu defines seven moments which are invariant to translation, rotation and scaling. These moments are calculated from central moments. Hu's seven moments are calculated according to the following equations:

Hu's moments are calculated from a normalised face image. Therefore, feature vector of size 7 is formed (Hu's seven moments $\phi 1$ to $\phi 7$). These feature vectors will be used to classify a face image [31].

C. Feature vectors and recognition

Feature vectors

Moment invariants methods, which were mentioned earlier, can be used as a feature extraction method in face images. Feature vectors can be obtained by calculating different orders of any moment invariant method. Generally, a feature vector can be written as:

Are moments of order". Usually, each order contains more than one moment. Therefore, calculating the feature vector for each image in a database would give a feature space matrix of size M x n, where M is the number of training images and n is the number of selected moment invariants for each moment invariant method [28, p.664].

Recognition

Having feature vectors for all training and test face images, the next step is to use a suitable classifier to recognise a face image. Different classifiers have been used such as nearest neighbour classifiers and neural network classifiers. Each classifier is considered to be suitable to classify specific types of feature vectors based on their characteristic. The nearest neighbour classifier is one of the widely used classifiers. It is based on comparing the feature vectors of training images in a database and the feature vector of a test image. This is found by calculating the Euclidean distance between training images and a test image. For example, are the classes in a database, the image is recognised by calculating the Euclidean distance between and k^{th} class and the feature vector with the minimum distance is considered to be the best match. (See equation **(26)**)([31].

26)

V. Implementation of Face Recognition Using Shape Descriptors (Hu's Seven Moment Invariants)

A. Working Procedure

To evaluate the performance of a face recognition algorithm using Hu's seven moment invariants, the algorithm is implemented with MATLAB scripts. The algorithm consists of two phases: training (learning) and testing (classification). These two phases work on my database.

Training phase

In this phase Hu's moments are used as a features extraction method to represent training images set in the database. Therefore, several steps should be performed. First of all, the training database path is selected. Then the ***Function[T, Train_Number] = Create Database(TrainDatabasePath)*** is invoked. This function gets one input argument, TrainDatabasePath, which represents the path of the training database. It returns a 2D matrix T, where each column represents a feature vector of an image in the training database. This feature vector is the Hu's seven moments of the image and of size 7. Therefore, the size of T is where 7 is Hu's seven moments of each image and 12 is the number of images in the training database. It also returns the Train_number argument, which represents the total number of images in the training database. In this function pre-processing steps are performed, imread and imresize commands are used to reduce the resolution of the images to 200 × 180 pixels and images are converted from colour images to binary images using the im2bw command. Inside the ***Create Database*** function a ***temp = feature_vec (img)*** function is invoked. This function gets the img argument, which is a 2D matrix representing an image and returns feature vectors of seven invariant moments of the image. Inside the ***feature_vec*** function another function ***cent_moment (p,q,A)*** is called to calculate the normalised central moments, where A is an input binary image and p, q are the order of moments.

Recognition phase

In the recognition phase, the Euclidean distance is used to compute the distance between feature vectors of training images and feature vector of test image. Minimum distance is considered to have a good recognition rate.

First of all, a test database path is selected. After that, a user enters the name of the test image, which is a number between 1 and 3. Then, ***Function OutputName = Recognition(TestImage, T, Train_Number)*** is invoked. The function gets the path of the test image, which is stored in the TestImage argument, T and Train_Number arguments (output from Create Database( )) and returns the OutputName of the corresponding image to the test image. In order to recognise a face image, several steps are performed. As in the training images database, firstly the test image is resized to pixels using imread and imresize commands in MATLAB and converted to a binary image. Then, to calculate the feature vector of the test image, the ***feature_vec (img)*** function is called – which is the same function mentioned in the training phase – to calculate Hu's seven moments of the test image. Finally, classification is done using the Euclidean distance classifier. Euclidean distances are calculated between feature vectors (Hu's seven moments) of training images and feature vector of test image. Minimum distance between test image and its match image in the training database is considered to have a good recognition rate.

Result and discussion

As shown in Figure 6, the Graphical User Interface (GUI) of the face recognition algorithm using Hu's seven moment invariants is designed using MATLAB. It consists of three pushbuttons: firstly, Hu's Moment Invariants button, which represents the training phase, which is mentioned in this chapter; secondly, Recognition Using Euclidean Distance button, which represents the recognition phase; and, thirdly, the Exit button. Simply, when the user clicks on the Hu's Moment Invariants button, the training phase is performed and training images are represented by their Hu's seven moments in the database. By clicking on the Recognition Using Euclidean Distance button, the recognition phase is done. Then, test and equivalent images as well as the name and age of the recognised person's face image are displayed. Finally, clicking on the Exit button terminates the program.

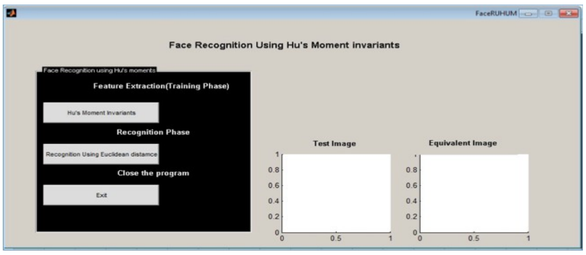

Figure 6. GUI of face recognition algorithm using Hu's seven moment invariants.

The result of the face recognition algorithm using Hu's seven moment invariants is shown in Figure 7. To recognise a person's face image, a user selects a test database path, then enters the number of the test image. Finally, the equivalent image of the testing image is determined by using Euclidean distance classifier

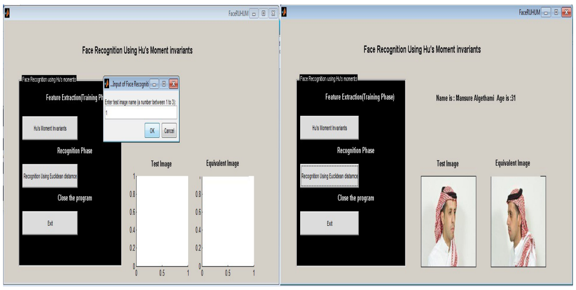

Figure 7. Result of face recognition algorithm using Hu's seven moment invariants

Performance analysis

As described in section III, for performance analysis of a face recognition algorithm using PCA as a feature extraction method, the same experiments were performed to evaluate performance of the algorithm using Hu's seven moment invariants.

The first experiment was performed on the face recognition algorithm using Hu's seven moments to determine the total number of training images needed to achieve the best recognition rate.

Observation: As the number of training images increases, the performance of the algorithm increased. The performance rate reaches the best recognition rate of 100% when the fourth training images are added.

The second experiment was performed to calculate the computation time of the training and testing phase of the face recognition algorithm using Hu's seven moments.

Observation: As in the PCA algorithm, computation time increases with increasing the number of training images in training and testing phases and computation time in the training phase is higher than in the testing phase.

No of the images in training database (per person)	Elapsed time in training phase (Seconds)	Elapsed time for each testing image (No. 1, 2 and 3)	Elapsed time in testing phase (Seconds) Elapsed time= $\frac{\textbf{Elapsed time for all testing images}}{\textit{the number of testing images}}$
1	13.8	No.1 = 10.8 No.2 =11.9 No.3 =11.1	33.8 / 3 = 11.3
2	20.6	No.1 =11.0 No.2 =12.3 No.3 =13.0	36.3 / 3 = 12.1
3	24.5	No.1 =12.4 No.2 = 12.4 No.3 = 13.1	37.9/ 3 = 12.6
4	28.5	No.1= 13.9 No.2 =13.3 No.3 = 13.9	41.1 / 3 = 13.7

VI. CONCLUSION AND COMPARISON OF THE TWO RESEARCH METHODOLOGIES, THE PRINCIPAL COMPONENT ANALYSIS AND HU'S SEVEN MOMENT INVARIANT.

A. Comparison Between Face Recognition Algorithm using a PCA Approach and using Hu's Seven Moments Approach

This section compares a proposed PCA-based face recognition algorithm with face recognition using Hu's seven moment invariants. These two feature extraction methods were performed on the same database (my database). In both algorithms, Euclidean distance classifier was used in the recognition phase. It is seen that using Hu's moments achieved the best recognition rate (100%) when the test images had different poses and lighting from those in the training database. This implies that Hu's moments as a feature extraction method provides features which are invariant to rotation, translation and scaling. While using the PCA method achieves 67.7% when the test images are different from the training images, PCA reaches 100% when the test images are presented in the training database.

Comparing the performance of two algorithms depends on the first and second experiments which have been done in performance analysis. In the first experiment, for both methods, the recognition rate increased as the number of training images increased; however, the number of training images that are required to achieve the best accuracy in PCA is five images, while in Hu's moments approach it is four. This is not a large difference, however, as the database is small. In the second experiment, for both methods, increasing the number of training images increased the computation time (in seconds) in both phases. It is also seen that the average computation time required to achieve the best recognition rate in Hu's seven moments method is 13.7seconds, while it is 14.1seconds in the PCA method. Training the algorithm to obtain a high recognition rate takes 28.5 seconds in Hu's seven moments method, while it is 25.1 seconds in the PCA algorithm.

B. Conclusion

In this paper, we have applied two feature extraction methods to a face recognition algorithm using my database. For the classification step Euclidean distances are used. These methods fall into two categories: a) holistic approaches and b) geometric feature-based approaches. Firstly, there is the Principal Component Analysis (PCA) method, which is an example of a holistic approach. It has been seen that large amounts of data can be reduced by using PCA. It is a mathematical algorithm which is simple to implement and use. The implementation of face recognition using the PCA approach has also been shown. As a result, when the number of training images and eigenvectors was increased, the recognition rate increased and this in turn increased the computation time. The algorithm achieves 100% when the test images are in the training database, while achieving 67.7% when test images are different from those in the training database.

The other feature extraction methods are shape descriptor methods, which are an example of feature-based face recognition. Shape descriptor methods fall into two groups: a) boundary-based shape descriptors and b) region-based shape descriptors. Moment invariants such as Hu's moment invariants are an example of region-based shape descriptors. It has been found that Hu's moments are invariant to translation, scaling and rotation. As a result of implementing face recognition using Hu's seven moments, the performance of the algorithm increased as the number of training images increased and computation time also increased. The algorithm reaches 100% recognition rate when the test images are different in pose and lighting conditions from training images in the database.

It was also found that using Hu's seven moments as features to represent images in the database is more robust to rotation than the PCA method. Moreover, average computation time using Hu's seven moments to get 100% recognition rate is 13.7 seconds, while in using the PCA method it is 14.1 seconds. Furthermore, it is seen that training time of the algorithm using Hu's moments is longer than using PCA, although the number of training images in the PCA algorithm is more than in Hu's seven moments method.

Finally, feature extraction is an important stage in the face recognition algorithm and the performance of the algorithm depends on the feature extraction method which is used in the algorithm. Moreover, many factors influence the performance of a face recognition algorithm such as database, face expression as well as pose and lighting conditions.

Given that the sample size is small, definite conclusion cannot be drawn from the results. If more time was permitted, the training set and test set would be increased in size, these giving more reliable results.

References

[1] P. Kamencay, D. Jelsovka, and M. Zachariasova, "The impact of segmentation on face recognition using the principal component analysis (PCA)," in *Proc. Signal Processing Algorithms, Architectures, Arrangements, and Applications (SPA),* 2011 [online] pp.1-4, 29-30. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6190931 &isnumber=6190920.

[2] D. Xue, Z. Li, and S. Hao, 'Face feature extraction and face recognition,' in: D. Jin, and S. Lin, *Advances in Electronic Engineering, Communication and Management Vol.1*. Berlin Heidelberg: Springer-Verlag, 2012, pp. 57-62.

[3] W.A Huang and H. Yin, "ViSOM dimensionality reduction in face recognition," in J. Principe and R. Miikkulainen, *Advances in Self-Organizing Maps: 7th International Workshop, WSOM* Germany: Spring-Verlag Berlin Heidelberg, 2009, p. 107.

[4] Z. Hafed, and M. Levine, "Face recognition using the discrete cosine transform," *International Journal of Computer Vision*, vol. 43, no. 3, pp. 167–188, 2001.

[5] M. Ballantyne, R. Boyer and L. Hines, "Woody Bledsoe: His life and legacy," *AI Magazine*, vol. 17, no. 1, pp. 7-20, 1996.

[6] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America A* vol. 4, no. 3, pp. 519-524, 1987.

[7] M. Ababneh, "Face recognition eigenface and fisherface performance," *European Journal of Scientific Research*, vol. 21, no. 1, pp. 6-12, 2008.

[8] S. Asadi, D. Rao, and V. Saikrishna, "A comparative study of face recognition with principal component analysis and cross-correlation technique," *International Journal of Computer Applications (0975 – 8887)*. vol. 10, no. 8, pp. 17-21, 2010.

[9] A. Eleyan and H. Demirel, "PCA and LDA-based face recognition using feedforward neural network classifier," in: B. Gunsel, A. Jain, A. Tekalp, and B. Sankur, *Multimedia Content Representation, Classification and Security*. Berlin Heidelberg: Springer-Verlag, 2006, 199-206.

[10] M. Abdullah, M. Wazzan, and S. Bo-saeed, "Optimizing face recognition using PCA," *International Journal of Artificial Intelligence & Applications (IJAIA)*, vol. 3, no. 2, pp. 23-31, March 2012.

[11] M. Dhanda (2012). Face recognition using eigenvectors from principal component analysis, *International Journal of Advanced Engineering Research and Studies* E-ISSN2249–8974 [Online]. Available:

International Journal of Advanced Engineering Research and Studies E-ISSN2249–8974

http://www.technicaljournalsonline.com/ijaers/VOL%20I/IJAERS%20VOL%20I%20ISSUE%20II%20JANUARY%20MARCH

%202012/IJAERS%2051.pdf Last accessed 5 Jun 2012.

[12] S. Zakariya, R. Ali, and M. Lone, "Automatic face recognition using multi-algorithmic approaches," in S. Aluru, S. Bandyopadhyay, U. Catalyurek, D. Dubhashi, P. Jones, M. Parashar and B. Schmidt, *Proc. Contemporary Computing: 4th International Conference, Ic3 2011, Noida, India, August 2011*. Berlin Heidelberg: Springer-Verlag, 2011, p. 501.

[13] T&T Laboratories Cambridge. (2002). *The Database of Faces.* Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

Last accessed 18th Sep 2012.

[14] Sandhu, P., Kaur, I., Verma, A., Jindal, S., Kaur, I. and Kumari, S. "Face recognition using eigen face coefficients and principal component analysis," *International Journal of Electrical and Electronics Engineering*, vol. 3, no. 8, pp. 498-502, 2009.

[15] Mahvish. (2012). *How Does PCA Eigenfaces Recognise a Face?* Available: http://fewtutorials.bravesites.com/entries/announcements/level-4b-part-ii-how- does-pca-eigenfaces-recognise-a-face-the-algorithm-steps. Last accessed 4th Aug 2012.

[16] Mathworks. (1984-2012). *Eigenvalues and Eigenvectors.* Available: http://www.mathworks.co.uk/help/techdoc/ref/eig.html. Last accessed 7th Jul 2012.

[17] W. Yambor, B. Draper, and J. Beveridge, "Analyzing PCA-based face recognition algorithms: eigenvector selection and distance measures," in: H. Christensen, and P. Phillips, *Empirical Evaluation Methods in Computer Vision*. London: World Scientific publishing Co. Pte. Ltd., 2002, pp. 39-54.

[18] L. Introna and H. Nissenbaum. (2009). *Facial Recognition Technology: A Survey of Policy and Implementation Issues.* Available: http://www.nyu.edu/ccpr/pubs/Niss_04.08.09.pdf. Last accessed 21st September 2012.

[19] V. Vaidehi, A. Annis Fathima, T. Treesa, M. Rajasekar, P. Balamurali, and M. Girish Chandra. (2011). *An Efficient Face Detection and Recognition System.* Available: http://www.iaeng.org/publication/IMECS2011/IMECS2011_pp499-503.pdf. Last accessed 28th September 2012.

[20] H. El-Bakry. "A rotation invariant algorithm for recognition," In: B. Reusch, *Computational Intelligence. Theory and Applications: International Conference, 7th Fuzzy Days*. Germany: Springer-Verlag Berlin Heidelberg, 2001, p. 284.

[21] C. Martinez-Ortiz. (2010). *2D and 3D Shape Descriptors.* Available: https://eric.exeter.ac.uk/repository/handle/10036/3026. PhD Thesis. Last accessed 28th September 2012.

[22] B. Kuhlenkoetter and X. Zhang. "A robot system for high quality belt grinding and polishing processes," in: V. Kordic, A. Lazinica and M. Merdan, *Cutting Edge Robotics*, Germany: The p IV pro literature Verlag Robert Mayer-Scholz, 2005, p.768.

[23] O. Sukmarg, D. Park, and K. Rao (2000). *Combined Shape Descriptors of Face Region and Contour of Human Object in MPEG Compressed Domain.* Available: http://spib.ece.rice.edu/DSP2000/submission/DSP/papers/paper046/paper046.pd f. Last accessed 29th September 2012.

[24] C. Solomon and T. Breckon. *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. USA: John Wiley & Sons, Ltd, 2011.

[25] S. Ashaghathra. *Identification of Pecan Weevils Through Image Processing*, USA: ProQuest Information and learning Company, 2008, p.32.

[26] J. Edwards, K. Riley, and J. Eakins "A visual comparison of shape descriptors using multi-dimensional scaling," in N. Petkov and M.A. Westenberg, Eds, *Proc. Computer Analysis of Images and Patterns,* 10th International Conference, CAIP 2003 Groningen, The Netherlands, August 2003.

[27] D. Zhang, and G. Lu. "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, pp. 1-19, 2004.

[28] A. Nabatchian, E. Abdel-Raheem and M. Ahmadi. "Human face recognition using different moment invariants: a comparative study," *2008 Congress on Image and Signal Processing*. 3, pp. 661- 666, 2008.

[29] M. Hu. "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179-187, 1962.

[30] R. Prokop and A. Reeves. "A survey of moment-based techniques for unoccluded object representation and recognition," *CVGIP: Graphical Models and Image Processing*. vol. 54, no. 5, pp. 438-460, 1992.

[31] A. Saradha and S. Annadurai. "A hybrid feature extraction approach for face recognition systems," *ICGST International Journal on Graphics, Vision and Image Processing*, vol. 5, no. 5, pp. 23-30, 2005.

[32] M. Mahoor, M. Abdel-Mottaleb, and A. Ansari (2006) "Improved active shape model for facial feature extraction in color images," *Journal of Multimedia,* vol. 1, no. 4.

Nahlah S. Algethami is a lecturer in Saudi electronic University, Saudi Arabia. She completed her Bachelor Degree in "Computer Science from

Taif University, Saudi Arabia in 2009. She completed her Master degree in Applied Computing for technologists from Dublin Institute of Technology, Ireland in 2013. Her research interests is Digital image processing.