

REPUBLIC OF TURKEY
FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES



FACE DETECTION AND RECOGNITION SYSTEM
USING PRINCIPAL COMPONENT ANALYSIS

MASTER THESIS

SHERWAN ABDULSATAR ABDULLAH

Supervisor: Assoc. Prof. Dr. Burhan ERGEN

MAY - 2017

REPUBLIC OF TURKEY
FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**FACE DETECTION AND RECOGNITION SYSTEM USING PRINCIPAL COMPONENT
ANALYSIS**

MASTER THESIS
SHERWAN ABDULSATAR ABDULLAH
142129109

Submission Date to the Institute: 25/4/2017

Thesis Presentation Date: 18/5/2017

(Title of Juror, Name and L/Name and institution):

Sign

Supervisor: **Assoc. Prof. Dr. Burhan ERGEN**

/ Firat Uni. / Comp. Eng.



Other Jury Members: **Prof. Dr. Ali KARCI**

/ İnönü Uni. / Comp. Eng.



Other Jury Members: **Assist. Prof. Dr. Ahmet CINAR**

/ Firat Uni. / Comp. Eng.



May - 2017

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my supervisor [Asst. Prof. Dr. Burhan ERGEN] for the continuous support of my Master's Degree study and related research, for his patience, immense knowledge and most importantly motivation. His guidance helped me in all the time of research and writing of this paper. I appreciated your passion and the way you delivered the lessons. Thank you for the structure and consistency you demonstrated in each meeting. Your work experience and professionalism enhanced the lessons. The product of this research paper would not be possible without you!

MAY-2017

SHERWAN A. ABDULLAH

LIST OF CONTENTS

ACKNOWLEDGEMENT	I
LIST OF CONTENTS	III
LIST OF FIGURES.....	V
LIST OF TABLES.....	VI
OZET	VII
ABSTRACT	VIII
1. INTRODUCTION	1
1.1. Face Recognition Problem.....	1
1.2. Face Detection	3
1.3. Organization of the Study	4
2. LITERATURE REVIEW	5
3. METHODOLOGY	8
3.1. Problem Definition	8
3.2. Background of the Method	9
3.3. Principal Component Analysis	11
3.3.1. Finding the Principal Components	12
3.3.2. Decomposing Variance.....	15
3.4. Face Recognition	16
3.4.1. Face/No-Face Decision and Identification	18
3.4.2. Near the Face Space.....	21
3.4.3. Near a Face Class.....	21
3.4.4. Euclidean Distance	21
3.4.5. Connection to the Bayes Classifier.....	22
3.5. Learning to Recognize New Faces	22
3.6. Locating and Detecting Faces.....	23
4. IMPLEMENTATION AND RESULTS	24
4.1. Preparing Database	24
4.2. Eigenface Implementation	25
4.2.1. The Training Set	26
4.2.2. The Testing Set	28
4.3. Eigenface Experimental Results	30
4.4. System Accuracy Calculation.....	32

5.	CONCLUSIONS.....	36
	REFERENCES	38
	CURRICULUM VITA	41



LIST OF FIGURES

Figure 1.1:	Three stages of the face recognition problem	2
Figure 3.1:	Different appearances of the faces	9
Figure 3.2:	The original image from a dummy training set of 20 images from the faces94 database.....	11
Figure 3.3:	Orthogonal transformation to the principal components.....	12
Figure 3.4:	Aim to minimise the distance between all points $\mathbf{x}$ and their projections $\mathbf{x}_0$. Image adapted from.....	13
Figure 3.5:	2 dimensional example with the two principal components plotted. Image adapted from.....	15
Figure 3.6:	Original image from the database not used in the training set and its projection into the face space using 11 Eigenfaces.....	17
Figure 3.7:	The weights associated with each of the 11 Eigenfaces in approximating the images in 3.6(b).....	17
Figure 3.8:	Face/No Face Decision flowchart	19
Figure 3.9:	Face Recognition Flowchart.....	20
Figure 4.1:	Sample of FERRT database faces.	24
Figure 4.2:	MY_DATABASE screen shot	25
Figure 4.3:	Mean face image	26
Figure 4.4:	Eigenfaces calculated from the MY_DATABASE training set.....	28
Figure 4.5:	Input face image with approximation image from MY_DATABASE	29
Figure 4.6:	Euclidean distance between the image and the face classes of each of the 11 individuals in the training set.	30
Figure 4.7:	Face detection and recognition images (a) individual training images (b) Tested image result after executing.....	33

LIST OF TABLES

Table 2.1: Results for Eigenfaces algorithm for real time face recognition.....	6
Table 4.1: Testing training set with increasing in individual images	34
Table 4.2: Summary of successful detection and recognition results.....	34
Table 4.3: Percentage successes ratio of system	34



OZET

Temel Bileşen Analizi Kullanarak Yüz Algılama ve Tanınma Sistemi

İnsan-makine etkileşimi, insan yeteneklerini taklit ederek insanlara bilgisayar desteğini arttırmaya odaklanan bir araştırma alanıdır. Bilgisayarlar, günlük yaşam görevinden başlayarak çok yetenekli iş görevlerine kadar destek sağlayabilir. Böyle bir göreve iyi bir örnek, bilgisayarlar tarafından daha verimli bir şekilde uygulanabilen otomatik yüz tanıma yöntemidir.

Bu tez, Eigenfaces tekniği ile PCA (Temel Bileşen Analizi) algoritmasını kullanarak yüz tanıma ve algılama sistemi sunmaktadır. Özgeçmişler, yüz tanıma için olağanüstü bir özellik olarak kabul edilen hızlı ve basit bir yöntemdir. Bu tezde önerilen yaklaşım, farklı baş, yüz profili ve kafa skalasında yüz tanıma sonuçlarını geliştirmek için algılama ve izleme için Eigenface yaklaşımı ile PCA tabanlı tekniği uygulamaktır. Bu, her birey için yalnızca beş resim içeren bir veri kümesi ile uygulanacak; görüntüler, yüzü bulmak ve resimdeki kişiyi tanımak için sınırlandırılmamış arka planın algılamanın karmaşıklığını artırdığı herhangi bir görüntü için farklı ortamlar için kısıtlanmamış bir arka plan olacaktır. Gerçek zamanlı uygulamalar için bu durumun çözülmesi gerekir.

Sonuçlar, eğitim görüntülerinin sayısının artmasının sistemin doğruluğunu artırabileceğini göstermiştir. Her bir bireye göre eğitim görüntülerinin sayısı, her alt grup için beş, bu yaklaşım için en uygun seçim eğitimi olarak bulunmuştur. Önerilen yöntemin varsayımını doğrulamak için standart veri kümesiyle karşılaştırma yapılmıştır; Sonuçlar algılama ve tanıma açısından iyi bir doğrulukla geliştirilmiş çalışma için Kabul edilebilir bir doğruluktur.

Anahtar kelimeler: Eigenfaces, yüz algılama, yüz tanıma, Temel bileşen analizi.

ABSTRACT

Man machine interaction is a research area that has been focusing on enhancing computer support to humans by mimicking their human abilities. Computers can provide support starting from the daily life task to highly skilled job tasks. A good example for such a task is automated face recognition that could more efficiently be performed by computers.

This thesis presents face detection and recognition system using PCA (Principal Component Analysis) algorithm with Eigenfaces technique. Eigenfaces is a fast and simple method for face recognition which is considered as a remarkable feature for it. Proposed approach in this thesis intent to implement PCA based technique with Eigenface approach for detection and tracking to enhance the results of face recognition in different head position, face profile and head scale. This will be implemented with a data set that have only five images for each individual, images will have unconstrained background for different environments for any given image to find the face and recognize the person in the image, where unconstrained background increase the complexity of the detection and recognition process, that requires these condition must be solved for real time implementation.

Results have been showed that increasing number of training images can improve accuracy of the system. The number of training images per each individual, which is five for each sub-set group found as the optimal choice training for this approach. Comparison with standard dataset of faces has been done for verifying proposed work assumption; results were fair and approving for developed work with good accuracy of detection and recognition.

Keywords: Eigenfaces, Face Detection, Face Recognition, Principal Component Analysis.

1. INTRODUCTION

Face recognition is one of the abilities that people develop in their early years, starting from the early months after their birth and keeps on improving with age. Some of us are very good at it and have an almost photographic memory while the others are not that good. The ability and performance of humans to recognize faces varies very much depending on many factors like memory, IQ, age, environment, face features etc. How are humans able to conduct this activity so effortlessly, yet the task of face recognition still remains one of the most challenging computer vision problems to date.

Over past years of research in this area have introduced many theories and methods on how face recognition can be implemented using a computer system.

The detection and recognition systems usually need a pre-defined database of known faces. However, if a new face has been considered the system has to decide whether it is a known face or not.

Using the pre-captured images database, the possibility of implementing a face recognition system based on the PCA Eigenfaces approach [1] is evaluated in this thesis. The focus lay on studying the algorithm, test and verify its performance with the database using a test setup especially created for this proposes. The results found during the evaluation were promising enough to use the algorithm as a basis to design and implement a face recognition system. Identifying the strong points and weaknesses of the algorithm produced a clear view on how further improvements and future work could be conducted to improve the system's performance.

1.1. Face Recognition Problem

The problem of face recognition can be simply stated as: "given an image of a scene identify or verify the identity of the face of one or more individuals in the scene from a database of known individuals" [1].

As shown in figure 1.1, the problem can be broken down into three distinct stages:

1. **Face detection and image pre-processing:** Faces need to be detected within a given scene and the image cropped to give a smaller image containing just the detected face. Early approaches to face detection focused on single face

segmentation using methods such as whole-face templates and skin colour, whilst later developments have led to automatic computer based face detection procedures. Once segmented the face image needs to undergo certain image processing methods to be roughly aligned and normalized. This aims to take into account factors such as lighting, position, scale and rotation of the face within the image plane [20].

2. **Feature Extraction:** Key facial features need to be extracted to enable identification of the face. Various approaches exist including holistic methods, such as Principal Component Analysis [5] and Linear Discriminant Analysis [20] which use the whole face and are based on the pixels' intensity values, and feature-based methods that identify local features of the such as eyes and mouth [3].

3. **Face Identification:** Based on the features found from feature extraction, this step attempts to classify the unknown face image as one of the individuals known by the machine. Various approaches for identification have been developed including many Eigenspace based approaches [21].



Figure 1.1: Three stages of the face recognition problem

This thesis going along all three stages of the problem, first the designed system detect face from selected image then it take it to other to stages namely that of feature extraction and then subsequent face identification based on these features. Furthermore, this image has then been segmented to give a smaller image just containing the detected face which then undergoes specified image preprocessing. We shall impose strict conditions on the images under investigation in that they are the entire same dimension.

1.2. Face Detection

Corner stone for all face recognition techniques is the face detection technique used, face detection purpose can simply described by finding the human's face in a given image from any source of photos used to capture them: still photos from database or digital camera that provide the system with still images or video camera as input stream. Finding humans face in a given photo and separate or cut the face part from the original image has specific steps to be done, starting with dimension reduction for data of face image because face has complex structure, and image may have complex back ground, then analyze the image to locate a face to determine face structure to be detected. Each detection technique has its own methodology, restrictions and determents in searching for the face features that may be not suitable for other ones that make the comparison between different algorithms limited, where no standards for testing the algorithms or evaluating their results are available, on the other hand this variation enrich recognition field and forces competition in results percentage for researchers. Even the calcification for these techniques varies according to the researcher's point of view in their study [3].

Researchers have different ways in classifying face detection approaches, some of them think that calcification of face detection methodologies falls in four categories of detection approaches according to the way these approaches study the structure of the face and locate these features to determine if the given image has a face in it or not with pointing to their cons and pros [4], [8]. Different Approaches of face detection calcification:

1. Knowledge-based methods: where these methods studies the patterns shape of faces that gives the information of the face structure component and the relation between these components.
2. Feature invariant or feature based approaches: this approach try to find facial features which are invariant to pose, lighting condition or rotation. Studying the face geometry by extracting the face features, these structural components are tested in different conditions of illumination and position factors which affect detection and recognition results, feature based is relatively fast in detection because it require low memory space than other approaches.

3. Template matching: one of the simplest approaches for face detection and recognition, this approach follows saved templates for facial features structure, calculate the correlation between a test image and pre-selected facial templates , maybe use one template or more for better results, where one template maybe considered insufficient which needs high memory resources.
4. Appearance-based methods: unlike the template matching this approach methodology is to teach the system using training sets of face images to help it to find the facial features structure, relatively fast in detection and recognition with higher accuracy than template matching, Eigenfaces subject of this thesis is following this category.

1.3. Organization of the Study

This report is structured as follows: Chapter One introduces the problem of face recognition and its many applications in the modern day world. Chapter Two shows literature review of several works related to face recognition area. Chapter Three introduces the method of Principal Component Analysis and its application with regards to face recognition, namely the method of Eigenfaces. The Eigenface method is then implemented with the results and subsequent limitations of the procedure discussed in chapter Four. Chapter five then concludes this report and gives a discussion of areas of potential interest for future work.

2. LITERATURE REVIEW

Face detection and recognition had a huge number of researches due to its important applications in all fields of human life. So it's impossible to review all literatures related to it. So in this chapter the focus will be on the main research methodologies which are most related to the proposed work.

In 1966, the first attempt to construct a semi-automated face recognition human-computer system was made [9], [10]. The system was based on the extraction of the coordinates of a set of features from the photographs, which were then used by the computer for recognition. Later, feature extraction and pattern classification techniques [14] were employed for face recognition purposes. In [7] and [18], a template matching approach was developed and improved, using automatic feature measurements and deformable templates which are parameterized models of face.

Early 1990s have witnessed the beginning of a new wave of developments for face recognition, with considerable research endeavors made for enhancing recognition performance. These include principal component analysis (PCA) [6].

Eigenfaces is classified as image base approaches, where its goal is to find out the eigenvectors of the covariance matrix of the distribution, the eigenvectors resulted from projecting training set of face images into linear subspace with lower dimensional space where the dimensionality reduction is achieved this subspace is the implementation of PCA concept on the face image appearance, Eigenfaces doesn't analyze the face feature variation [11].

Considered Eigenfaces approach as one of the best solutions that have been developed up-to-date, using dimensionality reduction for faces by considering each face as a vector in the face space where vector comparison is much easier and faster than matrices comparison [12]. The training set of face images where every face is represented by 2 dimensional matrixes is now a matrix with every face image is a vector; this vector is the face image after projecting to face space. This train set has constraints on the face image must be the same size to be calculated and to be frontal view to line-up most important facial features: eyes, mouth and nose these face features are analyzed and transferred into unrelated components known as orthogonal component named by Eigenfaces Arora, [17], another constraint lighting where image background is the wall or any object behind the

face that affect the recognition, must be aware of is the background where PCA face recognition is sensitive to the changes in these factors [19].

Implemented Eigenfaces recognition algorithm for real time face recognition using laptop computer and web camera, Arora considered video frames as still picture, his experiment examined changing or variation in illumination and head size, he considered the experiment quite successful [17]. Results of Arora's experiment where registered as following:

Table 2.1: Results for Eigenfaces algorithm for real time face recognition [17]

Face Condition	Recognition Accuracy	Recognition Error
Normal	83%	17%
Light Variation	61%	39%
Size of Face Variation	55%	45%

From the previous table results shows that Eigenfaces method is stable with fair recognition rate in normal condition (frontal head view, normal light condition, fixed head size), but unfortunately recognition rate drops when light conditions changed and the drop became worse when the scale vary which is bad impact on this technique. Studied the relationship between Eigenface recognition performance and different training data sets [20]. Using the Multilevel Dominant Eigenvector Estimation (MDEE) method, they were able to compute Eigenfaces from a large number of training samples. They focus more on the results of short feature lengths since they illustrate how efficient the transformation compresses the large face vector. As the length of the feature vector increases, it becomes more like the original face vector. The effect of the transformation is largely lost if the original face image directly was used for face recognition, they got an accuracy of 74.9%; their experimental results show that increasing the number of people benefits the recognition performance more than increasing the number of images per person. The gallery used for their research contains 72*10 face images of 72 different persons.

Face recognition systems gained high rate of accuracy when the learning set is large where each person have fair number of images that satisfies different head positions and facial expressions for better recognition results. Eigenfaces approach recognition

results are affected by the training data set images condition where unwanted (unconstrained) face image can defect the whole set recognition results [22].

In their research tried to minimize the participated eigenvectors which consequently decreases the computational time [1]. They conduct a study to optimize the time complexity of PCA (Eigenfaces) that does not affects the recognition performance. Their algorithm was tested on standard dataset: face94 face database experiments conducted using MatLab.

They conduct three experiments: The first experiment is used to adjust the best number of images for each individual to be used in the training set that gives a highest percentage of recognition. They choose 19 individual with 6 images for each in the training because the result of the first experiment shows that this number of images gives 100% recognition.

Second experiment tested 28 persons in the test database with 6 images for each person in the training database as given by experiment one. They changed the threshold trying to make a decision of the best matching. In this experiment they reduce the Eigenfaces for the PCA algorithm where eigenvalues are sorted and those who are less than a specified threshold are eliminated. Third experiment decreased the number of eigenvectors and consequently this decrease the time of computation. The results of this experiment give the same recognition result as the second experiment but with less time. The recognition time is reduced by 35%.

Said that people from different raises have different skin color nature, several studies have been done and found that the major difference lays in the variation between their skin color intensity rather than their chrominance [23].

Defined skin color based detection by: technique that used to separate skin pixels from the rest of colors in a given image, this technique is simple and requires less computation, but it is difficult to locate face in the presence of complex background and poor illuminations [24].

3. METHODOLOGY

This chapter has detailed view for the proposed approach for Eigenfaces using PCA for face recognition using real data with low restriction on training and testing data to simulate real environment.

3.1. Problem Definition

The performance of computer systems when performing face recognition also depends on many factors and is even more sensitive to changes, especially environment and input data changes. Figure 3.1 illustrates such changes. For these and other reasons performing face recognition should made in a specific controlled manner, based on a limited, properly defined input data. The face recognition problem of static images has been in general formulated as recognizing three-dimensional (3D) faces from two-dimensional (2D) images. The process of recognition depends on the implemented steps that could affect the results dramatically. The process starts first with capturing usable face images without missing essential face features, such as nose, eyes, eyebrows, mouth, and chin. The second step involves preprocessing the captured images to extract the face features that are essential for the recognition process, at the same time preprocessing discards (clips) unimportant parts of the image that could slow down or disturb the result. Depending on the used recognition algorithms, other steps such as training, classification, and finally recognition are triggered to recognize the individual presented to the system.

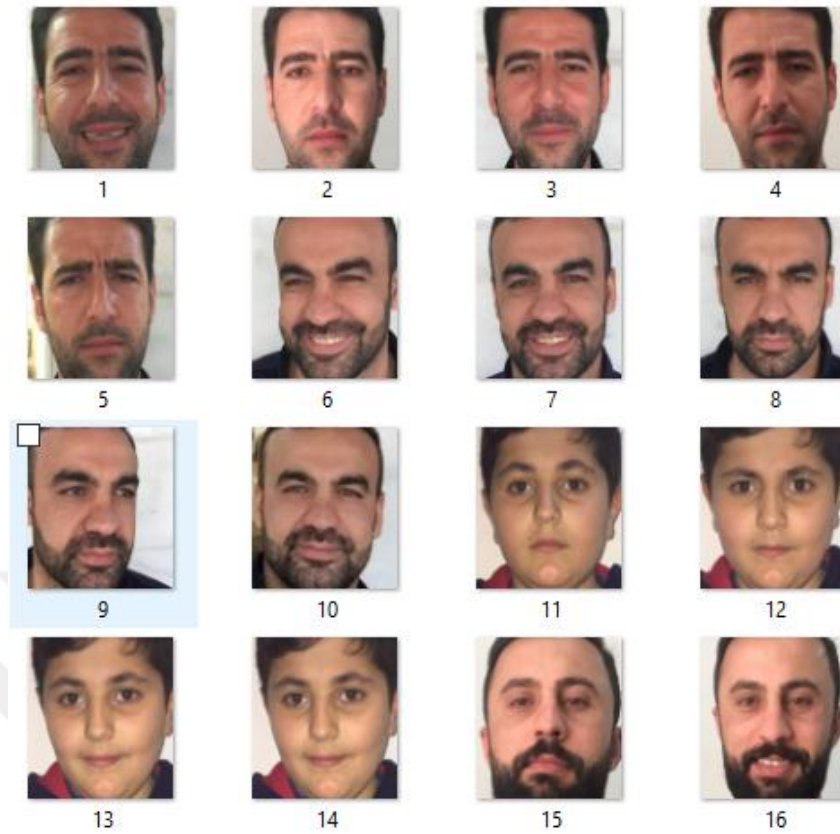


Figure 3.1: Different appearances of the faces

To develop an automatic system using existing technology, that will mimic the remarkable face recognition ability of humans with the advantage of a computer system and its capacity to handle large numbers of face images. Many studies in psychophysics and neuroscience have been conducted, which has delivered knowledge with direct relevance for engineers to developing algorithms in the domain of face recognition. This research focuses on evaluating the PCA Eigenfaces approaches. Using the evaluation results a system design that uses this algorithm shall be proposed to perform face recognition.

3.2. Background of the Method

The Eigenface method aims to reduce the dimensionality of the original image space by using Principal Component Analysis (PCA) to select a new set of uncorrelated variables. The aim is to choose these new variables in such a way that retains as much of the variation as possible from the original set of variables which define the original image space. This objective is then equivalent to finding the principal components of the image space. These principal components, also called Eigenfaces, can be thought of as a set of

feature vectors that represent the characteristic features within the face set and together this set of Eigenfaces characterize the variation between face images.

The core concept of the method is based on the observation that each face image can be exactly represented as a linear combination of the Eigenfaces. Furthermore, each face image can also be approximated using only the “*best*” Eigenfaces, observe figure 3.2. These “*best*” Eigenfaces are defined to be those accounting for the most variation within the set of face images. The Eigenfaces can then be thought of as a set of key features describing a set of face images. We then select the m best Eigenfaces, where $m \ll N$ the dimension of each image, to create the m -dimensional feature space which best represents the set of images. Every individual’s face can then be characterized by the weighted sum of the m Eigenfaces needed to approximately construct it in the feature space. These weights being stored in a feature vector of length m . This means that any collection of face images can be classified by storing a feature vector for each face image and a small set of m Eigenfaces. This subsequently greatly reduces the data storage requirements for storing a database of face images. Furthermore, this suggests that we are able to compare and recognize faces images by just comparing these feature vectors. This is the basis of the Eigenface method.

Using these low dimensional feature vectors we can determine two key facts about a face image in question: (If the image is a face at all?) If the feature vector of the image in question differs too much from the feature vectors of known face images (i.e. images which we know are of a face), it is likely the image is not a face.

If the image contains a known individual or not? Similar faces possess similar features (Eigenfaces) to similar degrees (weights). If we extract the feature vectors from all the images available then the images could be grouped into clusters, with each cluster representing a certain individual. That is, all images having similar feature vectors are likely to be similar faces. We can then determine if the image in question belongs to one of these individuals by its proximity to the clusters.

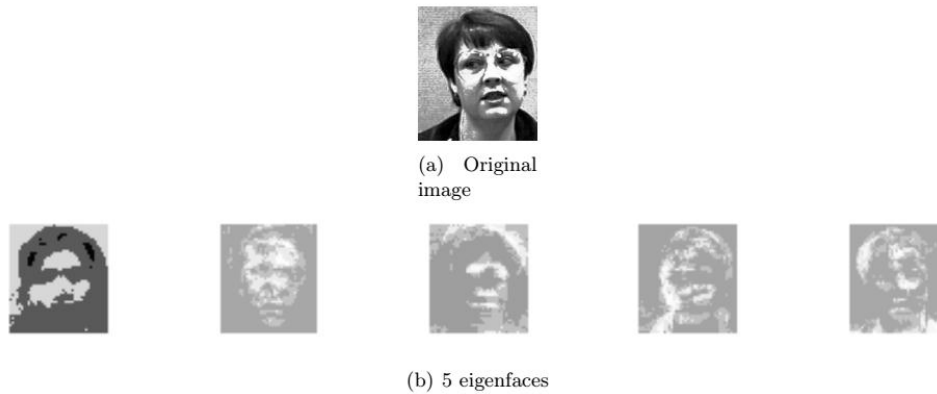


Figure 3.2: The original image from a dummy training set of 20 images from the faces94 database [1].

The Eigenface method can subsequently be seen in two steps:

1. A training set of images is used to find the Eigenfaces and hence train a computer to recognize the individuals in these images.
2. A testing set of images, i.e. a set containing images not in the training set, is considered to determine if the identity of the individuals in the images is known or not. We will begin by considering the training set and how to find the Eigenfaces using the concept of Principal Component Analysis.

3.3. Principal Component Analysis

As stated, Principal Component Analysis (PCA) aims to reduce the dimensionality of a data set consisting of a large number of potentially correlated variables, whilst retaining as much as possible of the total variation. This is achieved by an orthogonal transformation to a new set of uncorrelated variables, called the Principal Components (PCs) [15]. The first principal component is specified such that it accounts for as much of the variability in the original data variables as possible. Then each succeeding component in turn, is constructed to have the highest variance possible, under the constraint that it be orthogonal to the preceding components. PCA thus can be simply thought of as a coordinate rotation, aligning the transformed axes with the directions of maximal variance. This orthogonal transformation is shown in figure 3.3 which gives a plot of 50 observations on two highly correlated variables: $\mathbf{x}_1$ and $\mathbf{x}_2$ and a plot of the data transformed to the two PCs: $\mathbf{z}_1$ and $\mathbf{z}_2$. It is apparent that there is more variation in the

direction of $\mathbf{z}_1$ than either of the original variables $\mathbf{x}_1$ or $\mathbf{x}_2$. Also note there is very little variation in the second PC $\mathbf{z}_2$ which is orthogonal to $\mathbf{z}_1$.

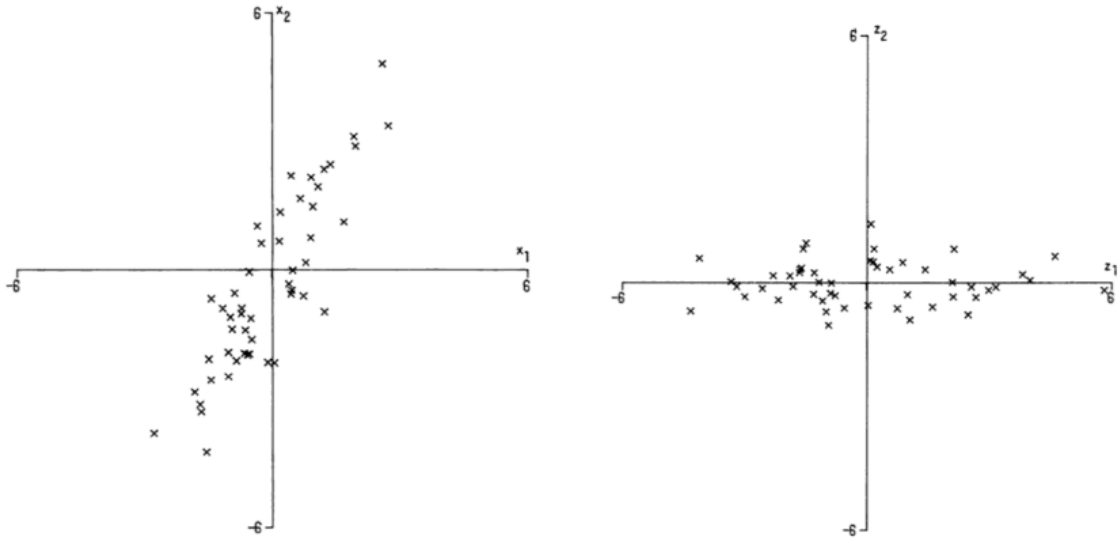


Figure 3.3: Orthogonal transformation to the principal components [15]

Plot of 50 observations on two variables x_1 and x_2 (b) Plot of 50 observations with respect to the two principal components $\mathbf{z}_1$ and $\mathbf{z}_2$

We can generalize this trivial example to a data set with more than 2 variables. If the variables have substantial correlations among them then it is hoped that the first few PCs will account for most of the variation in the original variables. This then suggests a dimension reduction scheme by transforming the original variables via a linear projection onto these first few PCs. We shall begin by considering how to find these PCs.

3.3.1. Finding the Principal Components

One way to examine the aim of PCA is to identify the linear directions in which the original data set is best represented in a least squares sense. We shall begin by considering the first principal component which is specified to be in the direction of maximal variability of the original variables. Thus in a least squares sense, the first PC is defined to be in the direction such that it minimizes the least squared error between the data points and their projection onto the line in this direction. Assume we have a data set consisting of M vectors each of length N . The mean vector of this data set is represented by μ and the covariance matrix is represented by Σ . The covariance matrix is defined to be the matrix whose $(i; j)^{\text{th}}$ element, $\Sigma_{i,j}$, is the covariance between the i^{th} and j^{th} elements of $\mathbf{x}$ when $i \neq j$

and the variance of the i^{th} element when $i = j$. Then consider, if we were to approximate x with a single straight line, which one would be the “best” line? This is then equivalent to selecting the direction in which there is maximal variability of the original data set. One approach to solve this problem is to look at it geometrically Pearson (1901) [13], Thus by Pythagoras we aim to:

$$\underbrace{E(\overline{xx'}^2)}_{\text{Min}} = E(\overline{\mu x'}^2) - \underbrace{E(\overline{\mu x}^2)}_{\text{Max}} \quad (3.1)$$

Where $E(\overline{xx'}^2)$ denotes the expected squared length of the line segment connecting x and x' , i.e. $\overline{xx'} = \|x - x'\|$. We take this expectation over all data points in the data set. Observe figure 3.4 for an illustration of this objective.

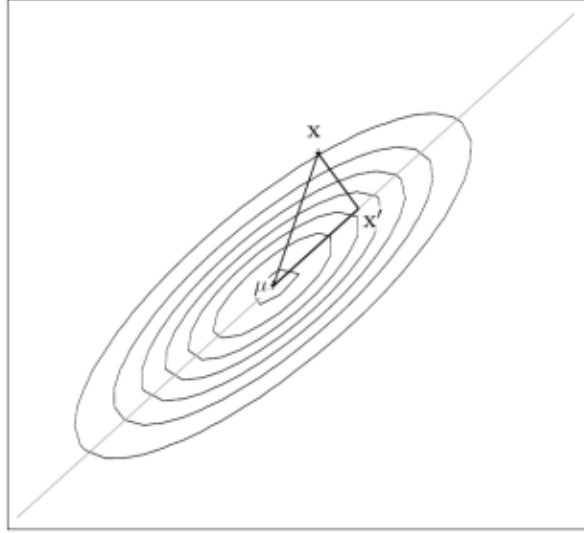


Figure 3.4: Aim to minimise the distance between all points x and their projections x' . Image adapted from [13].

As the distance $E(\overline{\mu x'}^2)$ does not depend on the fitted line, minimizing $E(\overline{xx'}^2)$ is equivalent to maximizing $E(\overline{\mu x}^2)$. We shall define γ to be the vector in the direction of the “best” line. Then investigating this quantity further:

$$E(\overline{\mu x'}^2) = E(\overline{\mu x'} \cdot \overline{\mu x'}) = E(\gamma^T (x - \mu) (x - \mu)^T \gamma) = \text{Var}(\gamma^T (x - \mu)) = \gamma^T \text{Var}(x - \mu) \gamma = \gamma^T \Sigma \gamma \quad (3.2)$$

Hence we need to maximize $\gamma^T \Sigma \gamma$. However it is clear that this maximisation will not be achieved for finite γ . We then need to impose a normalisation constraint. We shall

specify that $\gamma^T \gamma = 1$, that is γ to be of unit length. Note that other constraints can be used, but for simplicity in this derivation we shall use the unit vector constraint. This then gives a constrained maximization problem which can be addressed through the use of a Lagrange multiplier. Define:

$$P(\gamma) = \gamma^T \Sigma \gamma - \lambda (\gamma^T \gamma - 1) \quad (3.3)$$

Then the stationary points of above equation are the maximum values of $\gamma^T \Sigma \gamma$. Then:

$$\frac{dP(\gamma)}{d\gamma} = 2\Sigma \gamma - 2\lambda \gamma \quad (3.4)$$

Then setting this to zero yields:

$$\Sigma \gamma = \lambda \gamma \quad (3.5)$$

This is then an eigenvalue problem, thus the stationary values of $P(\gamma)$ are given by the eigenvectors of the covariance matrix Σ . We need to select the eigenvector that maximizes $P(\gamma)$. Note that $\Sigma \in \mathbb{R}^{N \times N}$ is a symmetric matrix so has N eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$ with associated orthonormal eigenvectors $\{\gamma_1, \dots, \gamma_N\}$. Observe that multiplying (3.5) by γ^T gives:

$$\gamma^T \Sigma \gamma = \lambda \gamma^T \gamma \quad (3.6)$$

$$\text{Var}(\gamma^T X) = \lambda \quad (3.7)$$

By using the fact that $\gamma^T = \gamma^{-1}$ as γ is an orthonormal eigenvector. Then the left hand side is now exactly what we want to maximize. Thus maximizing $\text{Var}(\gamma^T x)$ means then that we need to choose the eigenvector γ_1 corresponding to the largest eigenvalue, namely λ_1 . So $\gamma_1^T x$ is defined to first PC of $\mathbf{x} = (x_1, \dots, x_N)^T$. Thus the projection of $\mathbf{x}$ into the one-dimensional space given by the first PC is in the direction of maximal variance. Observe figure 3.4 which plots the two PCs of the example dataset. The second PC is specified such that it is in the direction of maximal variability subject to being orthogonal to the first PC. This is then given by the eigenvector with the second largest eigenvalue. In general, the k^{th} PC is given by $\gamma_k^T x$ with variance $\text{var}(\gamma_k^T x) = \lambda_k$, where λ_k is the k^{th} largest eigenvalue of Σ and γ_k is the corresponding eigenvector. As $\gamma_i \forall i$ are the orthogonal eigenvectors of Σ , this means the PCs give a set of orthonormal variables, as desired.

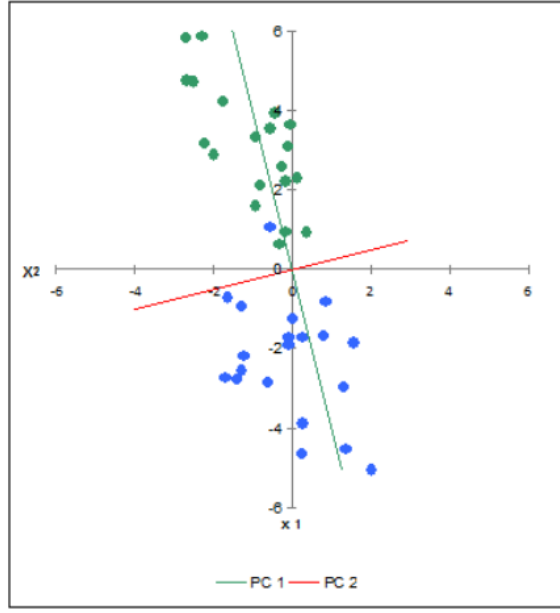


Figure 3.5: 2 dimensional example with the two principal components plotted. Image adapted from [13]

3.3.2. Decomposing Variance

The covariance matrix Σ is symmetric by its construction; hence it has a set of N orthogonal eigenvectors. From (3.5) and considering the j -th eigenvector γ_j of Σ we have:

$$\Sigma \gamma_j = \lambda_j \gamma_j \quad (3.8)$$

For each j , either side of this equation is a vector of length N . Then considering all N eigenvectors, we can bind these N vectors together to form an $N \times N$ matrix:

$$(\Sigma \gamma_1, \dots, \Sigma \gamma_N) = (\lambda_1 \gamma_1, \dots, \lambda_N \gamma_N) \quad (3.9)$$

Expressed in matrix form:

$$\Sigma(\gamma_1, \dots, \gamma_N) = (\gamma_1, \dots, \gamma_N) \begin{pmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_N \end{pmatrix} \quad (3.10)$$

Then further defining:

$$\Gamma = (\gamma_1, \dots, \gamma_N) \quad \Lambda = \begin{pmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_N \end{pmatrix} \quad (3.11)$$

Where Γ is the orthogonal matrix containing the eigenvectors and Λ is a symmetric, diagonal matrix with the eigenvalues along the diagonal and zeros in all other positions. Hence we can express (3.12) as:

$$\Sigma T = T \Lambda \quad \Sigma = T \Lambda T^{-1} = T \Lambda T^T \quad (3.12)$$

Any symmetric matrix can be diagonalised by its orthonormal eigenvectors. Thus each λ_j provides some decomposition of the variance, as considering just the j^{th} element gives:

$$\lambda_j = \text{Var}(\gamma_j^T X) \quad j = 1, \dots, N \quad (3.13)$$

So computing the sum over all j :

$$\lambda_1 + \dots + \lambda_N = \text{Tr}(\Lambda) = \text{Tr}(T \Lambda T^T) = \text{Tr}(\Sigma) = \text{TV}(x) \quad (3.14)$$

The trace of the variance matrix is then equal to the total variance of the data set $\text{TV}(\mathbf{x})$. Therefore the proportion of the total variance explained by the j -th principal component is given by:

$$\frac{\lambda_j}{(\lambda_1 + \dots + \lambda_N)} = \frac{\text{Var}(\gamma_j^T X)}{\text{TV}(X)} \quad (3.15)$$

3.4. Face Recognition

We will now consider the second step of the Eigenface method in which we have an image belonging to the testing set, i.e. an unknown image not present in the training set. We aim to determine if the individual in this image is known or not, that is if it is an image of one of the g individuals in the training set. We first need to find the feature vector for the new image in question, x_j , by projecting it onto each of the m Eigenfaces:

$$y_j = W^T (X_j - \mu) \quad (3.16)$$

Where μ is the mean face image from all images in the training set. This feature vector, $y_j = (y_{j,1}, \dots, y_{j,m})^T$, is of length m and represents the weighted sum of the Eigenfaces needed to approximately reconstruct the face image x_j . Observe figure 3.9 which gives the input image and the Approximated image using $m = 11$ Eigenfaces. Further

figure 3.7 illustrates the contribution of each of the 11 Eigenface in approximating the individual given in image in figure 3.6(b).



(a)

Original Image



(b)

approximation using 11 Eignenfaces

Figure 3.6: Original image from the database not used in the training set and its projection into the face space using 11 Eigenfaces.

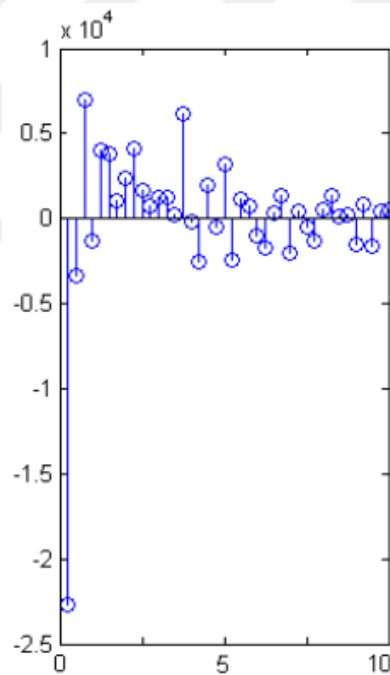


Figure 3.7: The weights associated with each of the 11 Eigenfaces in approximating the images in 3.6(b)

Once a new face image $\mathbf{x}_j$ has been projected into the face space and its feature vector $\mathbf{y}_j$ obtained, we can then determine:

1. If the image is of a face, whether known or unknown, i.e. is the image near the face space?

2. If the image is of a known or unknown individual, i.e. is the image near one of the defined face classes?

3.4.1. Face/No-Face Decision and Identification

The traditional Eigenface-based face recognition algorithm can be briefly described as an algorithm that extracts certain features from a given training set, and makes use of these extracted features to identify new faces given a testing set. The algorithm operates on full size face images and is generally used to identify a group of previously authorized people among a large set. The identification process begins when an unknown face is present at the system input. Firstly the system projects this new face image onto the face space and computes its distance from all the stored faces. The face can be identified as the individual that is nearest to the new projection in the face space. However since the projection onto the face space is a many-to-one mapping, several images looking nothing like a face could be projected onto a stored face vector. Therefore, before trying to identify the subject in a test image we need to decide if the presented test picture is a face image at all. We can decide if the input image is a face image by looking at the correlation (similarity measure) of the input face with the average face for the database in use. This process is depicted by Figure 3.8. If the computed correlation is equal or above a given threshold value T_c we assume we have a face image present at the input, otherwise we assume that the image present is not a face image.

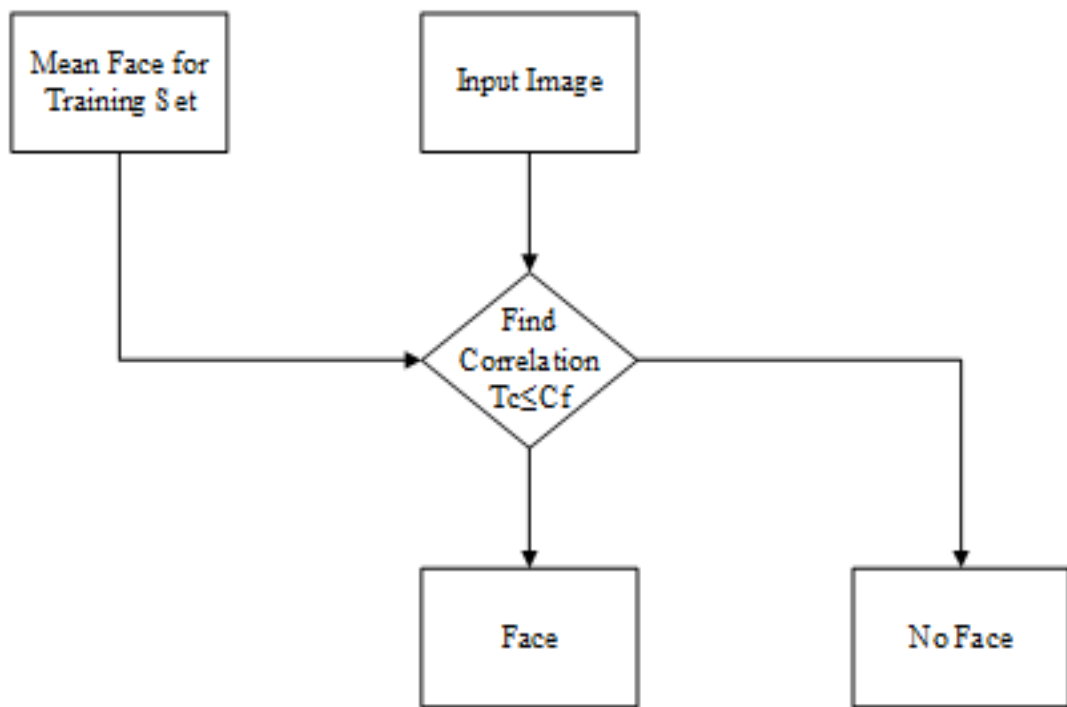


Figure 3.8: Face/No Face Decision flowchart

For normalized images the correlation C_f will be a value in the range 0.0 – 1.0. A value close to unity means that the input image is close to the mean face for the database and therefore it is highly likely that this is a face image belonging to the database. A more detailed flow diagram showing both the decision and identification processes are given in Figure 3.9.

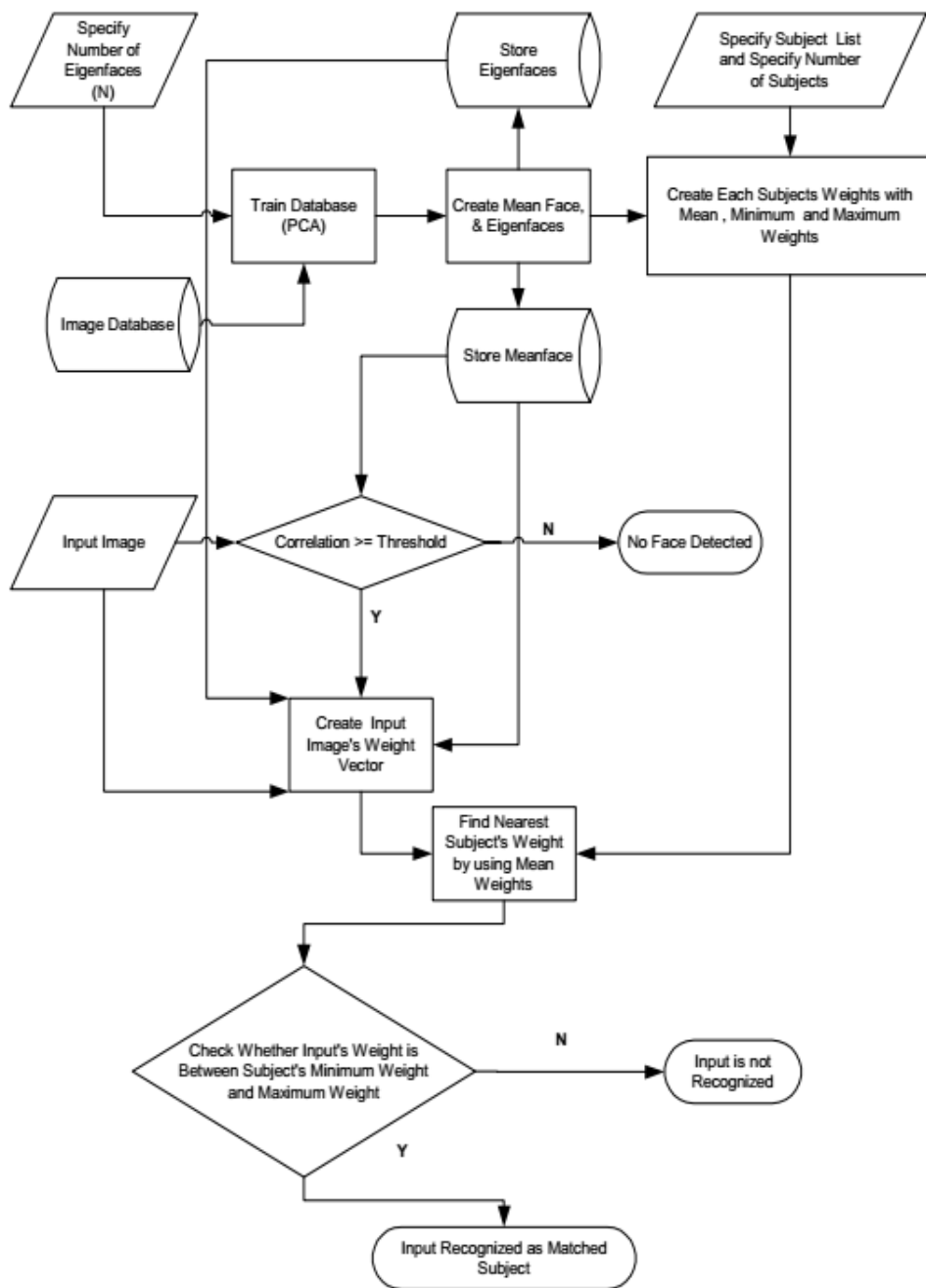


Figure 3.9: Face Recognition Flowchart

3.4.2. Near the Face Space

By projecting each face image into the low dimensional feature space it is likely that several images will project down onto the same feature vector. Not all of these images will necessarily look like a face. We thus need to use a measure of “*faceness*” to determine if the image is of a face or not. The distance between the image and the face space is a possible candidate for such a measure [5]. We shall consider using the Euclidean distance, which we will denote by. This is then just the difference between the mean adjusted images.

3.4.3. Near a Face Class

If the image in question is sufficiently close to the face space and hence classified as a face image, we can then look to determine if the image is of a known face.

3.4.4. Euclidean Distance

The nearest neighbor approach uses a distance similarity metric to identify a new face image by searching for the most similar vector in the training set to this new image’s feature vector. The simplest metric to use in such a method, as described in Turk and Pentland [5], is the Euclidean distance. This means we aim to find the face class k that minimizes the Euclidean distance between the feature vector $\mathbf{y}_j$ of the image in question and the average feature vector $\mathbf{y}^k$ of the k^{th} individual. This distance is given by:

$$\epsilon_k d(\mathbf{y}_j, \mathbf{y}_k') = \|\mathbf{y}_j - \mathbf{y}_k'\| = \sqrt{(\mathbf{y}_j - \mathbf{y}_k')^t (\mathbf{y}_j - \mathbf{y}_k')} \quad (3.17)$$

Formally this is nearest neighbor classification, as the image is classified by assigning it to the label of the closest point in the training set measuring all distances in the feature space. The face is determined to belong to class k if the distance is less than some predetermined threshold θ_k , that is:

$$\epsilon_k = \min_{i=1, \dots, g} \{\epsilon_i \mid \epsilon_i < 0_i\} \quad (3.18)$$

We need to specify a threshold θ_i for each of the $i = 1, \dots, g$ individuals. This then gives the maximum allowable distance the image in question can be from the average class vector of the k^{th} class to still be classified as the k^{th} individual. The image is classified as unknown if $\epsilon_i \geq \theta_i \forall i = 1, \dots, g$. Otherwise the image is classified as class k for which $\epsilon_k < \theta_k$ and the Euclidean distance k is the smallest distance over all possible distances, i.e. over all $\epsilon_i = \|y_j - y'_i\|$ for $i = 1, \dots, g$.

3.4.5. Connection to the Bayes Classifier

We shall take a slight detour and explore an interesting fact. Under certain normality assumptions on the training set, Bayes classification is equivalent to nearest neighbor classification using the Mahalanobis distance. Then under further assumptions this is also equivalent to using the Euclidean distance. It has been shown, [18], which the Bayes Classifier is the best classifier for identifying which class an unknown face belongs to in a parametric sense. This is because the Bayes Classifier yields the minimum error when the underlying probability density functions (PDFs) of each group are known, with this error is known as the Bayes error [16].

3.5. Learning to Recognize New Faces

If an image is sufficiently close to the face space but is distant from all known face classes, it is initially labeled as "unknown". However can we learn anything from these unknown face images? If a collection of "unknown" feature vectors cluster together this suggests the presence of a new but unidentified individual. The images corresponding to the feature vectors in this cluster can then be checked for similarity, by requiring that the distance from each image to the mean of the image is a less than a predefined threshold value. If the similarity test is passed this suggests all the images are of the same individual and thus a new face class can be added to the original training set. The Eigenfaces can then be recalculated to include these new face classes in addition to those classes in the initial training set to give $g + 1$ individuals we are now able to recognize.

3.6. Locating and Detecting Faces

We have just focused on the identification part of the face recognition problem when considering the Eigenface method. However we can utilize our knowledge of the face space to locate faces in an image, thus solving the first step of face detection of the problem. Faces do not tend to change by that much when projected into the face spaces whereas images of non faces do, [5]. Hence we can segment an image down into various subimages and then scan the image, at each location calculating the distance between the local subimage and the face space. This distance can then be used with a specified threshold to determine if the subimage is sufficiently close to the face space and thus if a face is present in the subimage or not. However note that this process of scanning the entire image is computationally extremely expensive.

4. IMPLEMENTATION AND RESULTS

4.1. Preparing Database

Most of the studies used standard faces database (data set) like FERET, Yale, ORL, MIT and many others that can be downloaded from the internet with referencing conditions enforced by data set providers. Researchers used these databases as training sets and test sets for faces images contain only the head in standard head scale meeting most needed face status in standard tests condition, they do this in order to eliminate the error rate for testing results due to face recognition limitations: illumination, face position, facial expression, facial occlusion (where part of the face is occluded by glasses, scarves or hats) and image background, which force restriction applies on the testing images to fit certain head size and certain background, where this kind of databases were prepared for this type of studies.



Figure 4.1: sample of FERET database faces [2].

In order to simulate the performance of PCA a large database is required. The results provided here are obtained using a database that has all been generated locally. This database contains single shots of people taken with a digital camera.

The database we have created was done so very naively. Originally we asked the family and friends to help us acquire different face pictures of selected people, with 5 poses for each individual. Out of the eleven subjects selected, two of the individuals were children. The face images were RGB type and their resolution was picked to be $(200 \times$

180). This database contains face images with different color backgrounds (even in the same set of five pictures of same individual) and has very poor illumination. Fortunately however, for almost all face images the size and location of the subject's head in the pictures was approximately same. Here in this database will be referred to as MY_DATABASE. Figure 4.2 below provides a snapshot of some of the pictures in MY_DATABASE.

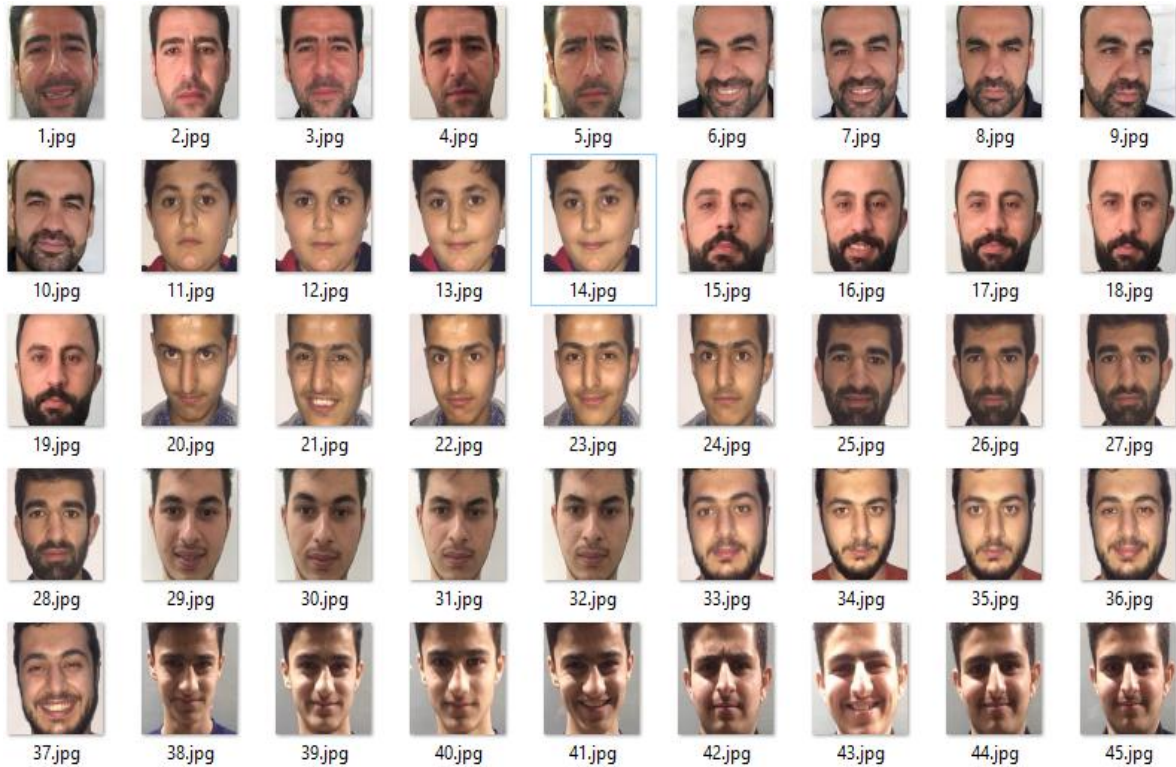


Figure 4.2: MY_DATABASE screen shot

Since the PCA algorithm operate on original face images with no pre-processing the background color or texture, or the size and location of a subject in the images can seriously affect the recognition performance of an automated system.

4.2. Eigenface Implementation

This section focuses on the computation details of training a computer system to implement and test the Eigenface method. A code written in MATLAB, to implement the method is given in **Appendix A** (using the Euclidean distance). The code covers both steps of the Eigenface method in first training a computer using the training set of face images and then secondly the identification procedure of classifying a non-trained face image. Both codes are identical apart from the distance measure used in classification. We shall

examine the code for each section and explain the computations in relation to the steps in the summary of the procedure given above.

4.2.1. The Training Set

Initially, a training set of face images is selected from a database to calibrate the machine. The method was implemented using MY_DATABASE and in each case a random selection of $g = 11$ individuals were selected from the database. $M_i = 5$ images of each individual were randomly chosen to form a training set of $M = 55$ images. Each image is of dimension 180×200 pixels. The images have all undergone image preprocessing of RGB to greyscale conversion using the MATLAB function *rgb2gray*. The first step of the code involves reading into the computer all 55 training images. Each image is first transformed from an array to a vector of length $N = 180 \times 200$ by concatenating each rows of the image array. We then use each of these vectors to form the $N \times M$ dimensional training matrix X , where each column $X(:,i)$ gives a the vector representing the i^{th} image.

- **Step One**

Involves calculating the mean face image of the training set. We use the data matrix X to calculate the mean value at each pixel position. Hence taking the mean across the i^{th} row of X to give the mean pixel value at the i^{th} position, to form the mean vector of length N . Observe figure 4.3 for the two mean images from the two training set considered.



Figure 4.3: Mean face image

- **Step Two**

Involves calculating the eigenvectors and eigenvalues of the covariance matrix $\Sigma = VV^T$. To reduce the amount of computation required of MATLAB in running code we calculate the eigenvalues of VV^T which is a 55×55 matrix as opposed to Σ which is of dimension 36000×36000 . We know the eigenvalues of Σ and VV^T are the same and the eigenvectors are related by $w_i = \frac{Vd_i}{\|Vd_i\|}$.

We further reduce the number of computations required by only considering eigenvectors which correspond to eigenvalues whose value is greater than zero. We thus form the vector L which contains all the non-zero eigenvalues i.e. $L_i > 0$ listed in descending order. The matrix D is then formed where the first column is the eigenvector of VV^T corresponding the largest non-zero eigenvalue and the last column is the eigenvector corresponding to the smallest non-zero eigenvalue. We then form the matrix L_eig_vec which contains the corresponding eigenvectors of Σ by multiplication of the matrix d by V . We specified that the eigenvectors are of unit length. Thus we normalize each column of the matrix U such that the eigenvectors are orthonormal i.e. $\|u_i\| = 1$.

Thus we have calculated the Eigenfaces and they are given by the columns of the matrix *Eigenfaces* in MATLAB (which is same U).

- **Step Three**

We need to specify the number of Eigenfaces, m , we are investigating. For the experimental procedures the code was run for $m = 10; 20; 30, 40$ Eigenfaces. Hence we select the first m columns of U to form the optimal $m \times N$ matrix W . We can then reshape each column of the matrix to visualize Eigenfaces.

Observe figure 4.4 which gives an examples of two such Eigenfaces. Note that in the first Eigenface the facial features such as the eyes, nose and hairline are visible whereas in the 40th Eigenface there is much more noise.



(a) first Eigenface



(b) 40th Eigenface

Figure 4.4: Eigenfaces calculated from the *MY_DATABASE* training set.

- **Step Four**

Covers the projection of each image in the training set into the face space, giving their corresponding feature vectors $y_i = W^T(x_i - \mu)$. We form a new $m \times M$ matrix *temp* which has each feature vector as a column. We then need to calculate the average feature vector for each individual. There are 5 images per individual in the training set so we need to take the average for each set of five consecutive columns of the matrix *temp*. This then gives a new $m \times 11$ matrix *Q*, where the k^{th} column gives the average feature vector for individual k , i.e.

$$y'_k = \frac{1}{5} \sum_{j=1}^5 y_j^k \quad (4.1)$$

4.2.2. The Testing Set

The second half of the code covers the face recognition part of the problem using a testing set to determine if the images in this set are of known individuals or not. The code in **Appendix A** uses the Euclidean distance as the distance measure. We shall focus on investigating the distance k of the untrained image to each of the 55 face classes as opposed to the distance to the face space.

This means the threshold θ , which gives the maximum allowable distance from the face space to be classified as a face image, is assumed to be infinite. Subsequently we assume, a priori, that all images are of faces. We shall construct 3 testing sets consisting of untrained images of the individuals in the training set, with each set containing one

randomly selected image of each of the $g = 11$ individuals. Thus each testing set contains 55 face images. We consider one set of these testing images at a time and run the second part of the code using the Euclidean distance, given the predetermined thresholds and a specified number of Eigenfaces m . Each image in the testing set is read in separately and the image's array turned into a vector of length N by concatenating the rows of the array. Then each image in the testing set is used a column vector to form the matrix S . The following procedure is done for the Euclidean distance from a differentiation in the final part of the code on how the distance between the feature vector in question and each of the face classes is specified.

- **Step Five**

Considering one image in the testing set at a time. This step covers the projection of the new image onto the m Eigenfaces to give its associated feature vector $y_k = W^T(x_k - \mu)$. We can then reshape this feature vector to observe the approximation of the image using the m Eigenfaces. Observe the approximations in figure 4.5 from MY_DATABASE.



(a)

Input face Image



(b)

Approximation using 11 Eignenfaces

Figure 4.5: Input face image with approximation image from MY_DATABASE

We then use a similarity metric of the Euclidean distance to measure the proximity of the image's feature vector to each face class, with the k^{th} face class given by $y'_s = Q(:, k)$. Observe figure 4.6 which illustrates the Euclidean distance to each of the 11 face classes.

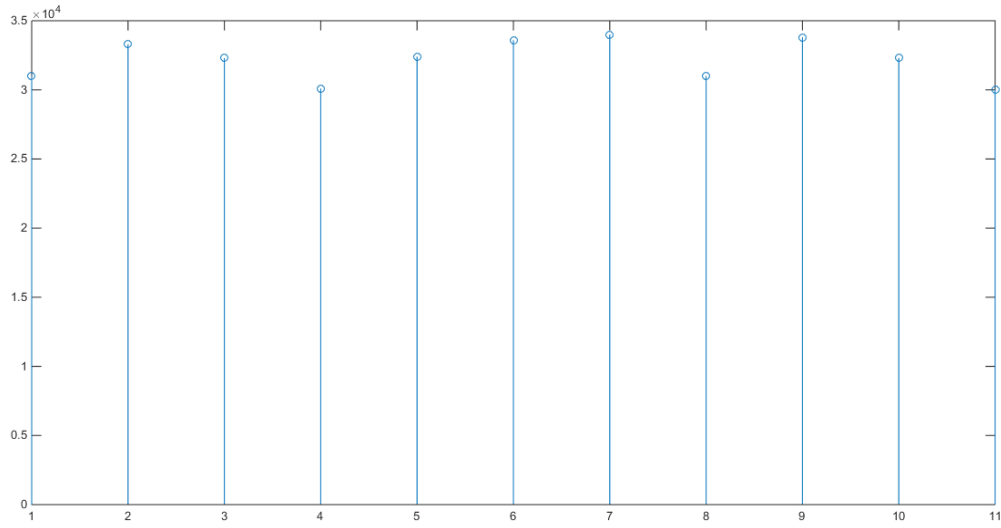


Figure 4.6: Euclidean distance between the image and the face classes of each of the 11 individuals in the training set.

4.3. Eigenface Experimental Results

In implementing the Eigenface method there are several variables that can either be varied or need to be kept constant. In investigating this method we shall examine the effect of 3 key variables:

- **Number of Eigenfaces**

We shall take $m = 10; 20; 30; 40$ and examine how this affects the accuracy of recognition of the method. Note that it is useful to know the optimum value for m as we seek to find the optimal balance between recognition accuracy of the method (i.e. retaining enough variation of the original image space to be able to accurately represent an image) and the compression of the image space to reduce data storage and computational requirements.

- **Distance Measure**

We shall investigate using the Euclidean distance in measuring the distance between the feature vector of each image in the testing set and each face class. We shall keep the threshold parameters $\theta_i \forall i = 1, \dots, g$ constant. We shall only consider a testing set in which the images are of individuals in the training set thus can assume a priori that all images are sufficiently close to the face space hence none should be rejected as non-face

images. We are subsequently only interested in the recognition accuracy of the method; hence the threshold θ is taken to be infinite.

- **Image Database**

We shall investigate using the *MY_DATABASE*. This database has different characteristics, in particular in terms of the amount of pose and lighting variation present between images. Implementation on this database will enable a comparison to see how well the Eigenface method is able to cope with such sources of variation.

Throughout the experimental procedure the same training set of $M = 55$ face images of $g = 11$ randomly selected individuals shall be used, one created from each database. We will also keep constant the allowable thresholds $\theta_i \forall i = 1, \dots, 11$. We shall experiment using faces database. This database will have its own associated 55 image training set and 3 sets of testing images. We shall run the code three times on each testing set to ensure the reliability of the results. We can then define the percentage recognition accuracy of the procedure to be the percentage of correctly classified individuals given a fixed threshold and specified distance measure. We shall take an average of this recognition accuracy for a given number of Eigenfaces and distance measure over the results from the 3 repeats for each of the 3 testing sets. This will ensure the reliability of the final results. We can further calculate the standard deviation (SD) for the recognition accuracy to give a measure of the reliability of the results produced from the method. This then enables an error bound to be calculated for the average recognition accuracy value stated. In experimenting we shall select the threshold $\theta_i \forall i = 1, \dots, 11$ in such a way to reduce the number of images not recognized as any of those within the database, i.e. to reduce the rejection rate of images.

This is because we know the testing sets are constructed from images of known individuals, i.e. individuals within the training set, so we know a priori that all images in the set are of known individuals.

We want to test if we have trained the computer sufficiently, using the Eigenface method, to be able to recognize these individuals. We are subsequently not so interested in the rejection rate of the Eigenface procedure but the recognition ability of the method. This means we need to select a high threshold, thus all images will be classified as their nearest

neighbor. However note that this is likely to give more errors than a smaller threshold would give. The thresholds are taken to be:

$$\text{Euclidean distance: } \theta_i = 1 \times 10^{-7} i = 1, \dots, 11 \quad (4.2)$$

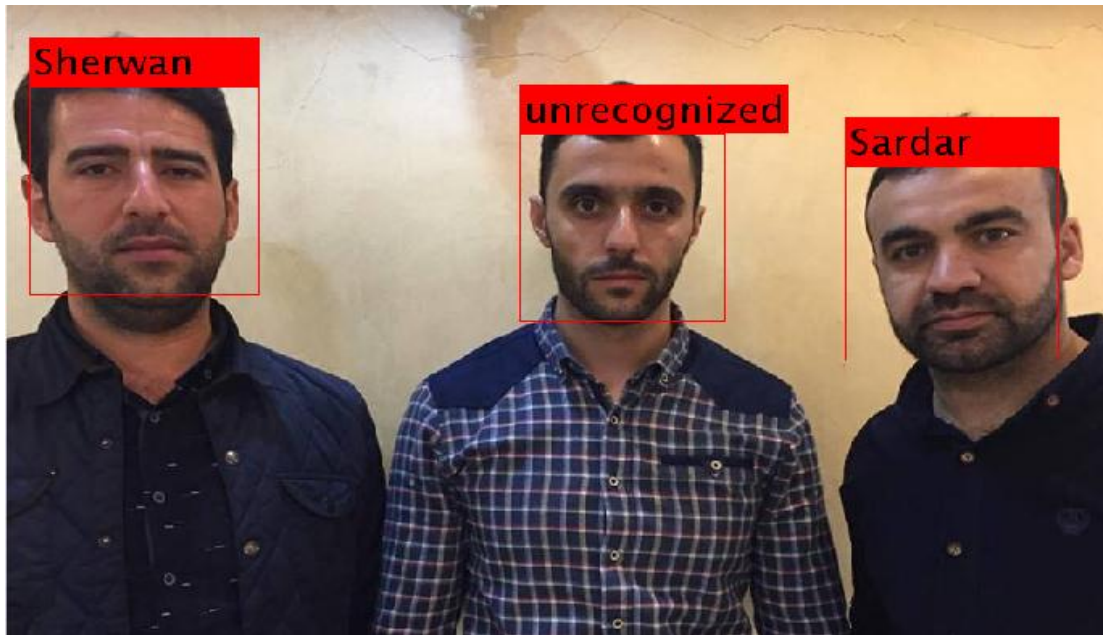
These thresholds mean that some images are rejected as unknown and each image is not classified as its nearest neighbour. Figure gives a visual example of this nearest neighbour classification using the specified threshold for the Euclidean distance.

4.4. System Accuracy Calculation

For testing and calculating the accuracy of the system, it have been runes several times to record the successes and failed data. Example is given for test on an image testing the accuracy to recognize individual with different view for the test image, the test scored true recognition. Figure 4.7 showing the equivalent image.



(a)



(b)

Figure 4.7: Face detection and recognition images (a) individual training images (b) Tested image result after executing

Then the test have been implemented in other phases as following: where the data set is available for 11 persons and dividing technique into sub-set for the training set is implemented, plus the aim of the proposed approach to decrease the number of the training images for each individual, this type of challenge need excessive testing.

This type of tests will find the effect of both factors:

First: decreasing the number of images for each individual, starting with minimum number of training images for each individual and increment this number (starting by 1 for each ending by 5) and observe the results.

Second: sub-setting training data set, in parallel with changing training images numbers sub-setting the training set will be tested also (starting by 1 individual in each subset ending by 11).

Detailed testing results are listed in table 4.1, tests have been categorized according to the number of training images for each individual, results of recognition are listed in tables with recognition result for each individual, correct recognition result is marked with (✓) sign.

Table 4.1: Testing training set with increasing in individual images

Image No.	Number of training images				
	1	2	3	4	5
1		✓	✓	✓	✓
2		✓			✓
3		✓	✓	✓	✓
4					
5				✓	✓
6	✓	✓	✓	✓	✓
7	✓	✓	✓	✓	✓
8				✓	✓
9	✓	✓	✓		✓
10			✓	✓	✓
11	✓			✓	✓

These results are followed by summarized table of results for easier observing recognition accuracy and analyzing which is shown in table 4.2.

Table 4.2: Summary of successful detection and recognition results

Result	Number of training images				
					5
Correct					10
False					1

Summary for this test for the obtained tables is represented with recognition percentage for each case and it is shown in table 4.3.

Table 4.3: Percentage successes ratio of system

Result	Number of training images				
	1	2	3	4	5
Recognized	36.4%	54.5%	54.5%	72.7%	90.9%
Unrecognized	63.6%	45.5%	45.5%	27.3%	9.1%

The results shows that by increasing the number of training images from 1 to 5 images for each individual increased the recognition results remarkably, but then from 2 to 3 it has less effect on the recognition. Although sub-setting the training data set solved the problem of having small number of training set for each person. The highest result from previous work with increasing both training images and the number of individuals in the sub-set group was (5 training images per each and 6 or 7 individual for each sub-set group) this is the optimal choice training for this approach.



5. CONCLUSIONS

In this report we have focused our efforts on the identification stage of the face recognition problem. We have considered Eigenfaces approach to determine the optimal way to extract discriminating features from a set of face images and then identify the individual in an image based on these extracted features. We began by considering linear subspace analysis to attempt this feature extraction. The idea was that we would be able to make the high dimensional image space more compact and useful of discriminatory purposes, by linearly projecting face images down into a new lower dimensional subspace, as defined by certain optimal features. We had begun by considering the Eigenface method, given in Chapter 3, in which we used PCA to define the low dimensional feature space.

The idea of PCA is to select a set of orthonormal basis eigenvectors, called the principal components, with the aim of retaining as much variability from the original space as possible. We called these principal components Eigenfaces. We showed how any face image can be approximately reconstructed using just a small number of these Eigenfaces. Furthermore, considering face database created for this work, our experimental analysis showed that using just the first 7 Eigenfaces was optimal to maximize the recognition accuracy of the Eigenface method.

However the success of the Eigenface method is inhibited by the 3 key limitations occurring due to the use of PCA. The first being the assumption that large variances are important. This means the Eigenface method selects the most expressive features, which could include those due to lighting and pose variations. We saw, through our experimental investigations, how such features are not optimal for discriminatory purposes. The second being due to the non-parametric nature of PCA. This is because PCA does not make any assumptions on the structure of the data set. This subsequently leads to data loss, as we do not account for class reparability. The third limitation being due to the linear nature of the method. In using a linear projection we are unable to take into account any of the higher order pixel correlations present in face images. The Eigenface method consequently assumes that such nonlinear relations are not necessary to discriminate between different individuals.

There are still numerous issues which must be addressed in order to ensure the robustness of a working face recognition system based on the methods we have discussed. We shall briefly mention two key areas and hint towards a potential solution that could be explored in future work.

In all preceding analysis we have implicitly assumed a Gaussian distribution of the face space when considering a nearest neighbour approach to classification. We have discussed how a Gaussian distribution appears reasonable. However, it is difficult if near not impossible to estimate a true distribution of face images. We subsequently have no a priori reason to assume a particular density function of face images. It would then be useful if we could develop an unsupervised method that enables us to learn about the distribution of face classes to either confirm our Gaussian assumption or propose an alternative model. Nonlinear networks suggest a useful and promising way to learn about the face space distributions.

REFERENCES

- [1] **Abdullah, M., Wazzan, M., Bo-saeed, S.** 2012. "Optimizing face recognition using PCA", International Journal of Artificial Intelligence Applications IJAIA, 32,23-31,.
- [2] **Geng, C., Jiang, X.** 2009. "SIFT Features for Face Recognition", IEEE. 598-602, 978-1-4244-4520-2/09.
- [3] **Hjelm, E., Low, B.K.,** 2001. "Face Detection: A Survey, Computer Vision and Image Understanding", 83, 236–274.
- [4] **Anila, S., Devarajan, N.** 2010. "Simple and Fast Face Detection System Based on Edges", International Journal of Universal Computer Sciences.1 2, 54-58.
- [5] **Belhumeur, P. N., Hespanha, J. P., Kriegman, D. J.** 1997, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection". IEEE Trans. Pattern Analysis and Machine Intelligence, 197, 711-720.
- [6] **Chakraborty, D.** 2012. "An Illumination invariant face detection based on human shape analysis and skin color information". Signal Image Processing: An International Journal SIPIJ, 3, 55- 62.
- [7] **Chellappa,R., Wilson,C.L., Sirohey, S.** 1995. "Human and machine recognition of faces: A survey", IEEE, 83 5, 705 - 741.
- [8] **Tin, H. H. K.** 2012. "Robust Algorithm for Face Detection in Color Images", I.J.Modern Education and Computer Science, 2, 31-37, 10.5815.
- [9] **Duda, R. O., Hart, P, E.** 1972."Use of the Hough Transformation To Detect Lines and Curves in Pictures", Communications of the ACM, 11-15.
- [10] **Gao, W., Yang L., Zhang, X., Liu, H.** 2010. "An Improved Sobel Edge Detection", IEEE, 67-71, 978-1-4244-5540-9/10.
- [11] **Turk, M., Pentland, A.** 1991. "Eigenfaces for recognition". Journal of Cognitive Neuroscence, 31, 71–86..
- [12] **Belhumeur, P. N., Hespanha, J. P., Kriegman, D. J.** 1997, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection". IEEE Trans. Pattern Analysis and Machine Intelligence, 711-720.
- [13] **Georgieva, L., Dimitrova,T., Angelov, N.** 2005. "RGB and HSV colour models in colour identification of digital traumas Images". International Conference on Computer Systems and Technologies – CompSysTech, 1-6.
- [14] **Gottumukkal, R., Asari, V.K.** 2004. "An improved face recognition technique based on modular PCA approach, Pattern Recognition Letters", 25,429–436,
- [15] **Hjelm, E., Low, B.K.,** 2001. "Face Detection: A Survey, Computer Vision and Image Understanding", 83, 236–274.
- [16] **Islam, A., Imran,S., Isalm, R.U., Abdus-Salam, M.** 2003. "A Multifarious Faces and Facial Features Detection Algorithm for Color Images Based on Grey-scale Information and Facial Geometry", ICIT.
- [17] **Arora, K.** 2012. "Real Time Application of Face Recognition concept", International Journal of Soft Computing and Engineering IJSCE, 2 5, 2231-2307.

- [18] **Kondo,T., Yan, H.**, 1998, "Automatic human face detection and recognition under nonuniform illumination, *Pattern Recognition* 32, 1707-1718.
- [19] **Slavkovic, M., Jevtic, D.** 2012. "Face Recognition Using Eigenface Approach". *Serbian Journal of Electrical Engineering*, 91, 121-130.
- [20] **Li, Z., Tang, X.** 2002." Eigenfaces recognition using different training data sizes". *CM*. 1-58113-369-3.
- [21] **Manyam, O.K., Kumar, N., Belhumeur, P., Kriegman, D.**, 2011, "Two faces are better than one: Face recognition in group photographs". *IEE*, 978-1-4577-1359-0/111.
- [22] **Ramesha, K.** 2009. "Advanced Biometric Identification on Face, Gender and Age Recognition", *IEEE journal*, 978-0-7695-3845-7/09, 10.1109.
- [23] **Zhang, Q., Zhang, J.** 2009. "RGB Color Analysis for Face Detection", *D M Akbar Hussain,Advance in computer and IT*,pp 109-125,.
- [24] **Thepade, S., Khandelwal, S., Dhamejani, K., Azmi, A.** 2012. "Improved Face Recognition with Multilevel BTC using Kekre's LUV Color Space", *IJACSA International Journal of Advanced Computer Science and Applications*, 31, 156-160.
- [25] **Abul-Kashem, M., Akhter, M. N., Ahmed, S., Alam, M. M.**, 2011, "Face Recognition System Based on Principal Component Analysis PCA with Back Propagation Neural Networks BPNN", *Canadian Journal on Image Processing and Computer Vision*. 24,36-45.
- [26] **Anila, S., Devarajan, N.** 2010. "Simple and Fast Face Detection System Based on Edges", *International Journal of Universal Computer Sciences*.1 2, 54-58.
- [27] **Chakraborty, D.** 2012. "An Illumination invariant face detection based on human shape analysis and skin color information". *Signal Image Processing: An International Journal SIPIJ*, 3, 55- 62.
- [28] **Georgescu, D.** 2011. "A Real-Time Face Recognition System Using Eigenfaces", *Journal of Mobile, Embedded and Distributed Systems*, 34, 2067 – 4074.
- [29] **Chitra,S., Balakrishnan, G.** 2012. "Comparative Study for Two Color Spaces HSCbCr and YCbCr in Skin Color Detection", *Applied Mathematical Sciences*, 685, 4229 – 4238.
- [30] **Duda, R. O., Hart, P,E.** 1972."Use of the Hough Transformation To Detect Lines and Curves in Pictures", *Communications of the ACM*, 11-15.
- [31] **Georgieva, L., Dimitrova, T., Angelov, N.** 2005. "RGB and HSV colour models in colour identification of digital traumas Images". *International Conference on Computer Systems and Technologies – CompSysTech*, 1-6.
- [32] **Gottumukkal, R., Asari, V.K.** 2004. "An improved face recognition technique based on modular PCA approach", *Pattern Recognition Letters*, 25, 429–436,
- [33] **Farzin Mokhtarian, Sadegh Abbasi and Josef Kittler**, 2008. "Robust and Efficient Shape Indexing through Curvature Scale Space", *Department of Electronic Electrical Engineering, University of Surrey, England*.
- [34] **Lu, Y., Zhou, J., Yu, Sh.** 2003. A survey of face detection, extraction and recognition, *Computing and Informatics*, 22, 163-195.

- [35] **Süsstrunk, S., Buckley R., Swen,S.** 1999. Standard RGB Color Spaces. Tathe,S. V., Narote, S. P. 2012. “Face detection using color models, World Journal of Science and Technology”, 24, 182-185.
- [36] **Yoo, S., Park, R.H., Sim, D.G.** 2007. “Investigation of Color Spaces for Face Recognition”. Conference on Machine Vision Applications, 106-109.





CURRICULUM VITA

SHERWAN ABDULSATAR ABDULLAH

Zakho – Iraq

sherwan_zaxoy@yahoo.com

PERSONAL DETAILS:

Birth Date:	27/10/1986
Birth Place:	Zakho- Iraq
Gender:	Male
Marital Status:	Married
Telephone number:	00964-750-4521966

EDUCATION:

2006 - 2010	B.Sc. at University Of Nawroz – Computer Science Department
2015 - 2017	MSc at University Of Firat - Computer Engineering Department

EXPERIENCE:

2012 - 2017	Lecturer and Research Assistant Computer Science Department University Of Zakho
-------------	---