A COMPARISON OF SUBSPACE BASED FACE RECOGNITION METHODS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖZKAN GÖNDER

IN PARTIAL FULLFILMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2004

Approval of the Graduate School of Natural and Applied Sciences.

_____

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Prof. Dr. Mübeccel DEMİREKLER
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Prof. Dr. Uğur HALICI
Supervisor

Examining Committee

Prof. Dr. Kemal LEBLEBİCİOĞLU	(METU,EE)	_____
Prof. Dr. Uğur HALICI	(METU,EE)	_____
Prof. Dr. Mete SEVERCAN	(METU,EE)	_____
Dr. Ece GÜRAN	(METU,EE)	_____
M.Sc. Alper ÜLKÜ	(ASELSAN)	_____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Özkan GÖNDER

Signature :

ABSTRACT

A COMPARISON OF SUBSPACE BASED FACE RECOGNITION METHODS

Gönder, Özkan
M.Sc., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Uğur Halıcı

August 2004, 95 pages

Different approaches to the face recognition are studied in this thesis. These approaches are PCA (Eigenface), Kernel Eigenface and Fisher LDA. Principal component analysis extracts the most important information contained in the face to construct a computational model that best describes the face. In Eigenface approach, variation between the face images are described by using a set of characteristic face images in order to find out the eigenvectors (Eigenfaces) of the covariance matrix of the distribution, spanned by a training set of face images. Then, every face image is represented by a linear combination of these eigenvectors. Recognition is implemented by projecting a new image into the face subspace spanned by the Eigenfaces and then classifying the face by comparing its position in face space with the positions of known individuals. In Kernel Eigenface method, non-linear mapping of input space is implemented before PCA in order to handle non-linearly embedded properties of images (i.e.

background differences, illumination changes, and facial expressions etc.). In Fisher LDA, LDA is applied after PCA to increase the discrimination between classes.

These methods are implemented on three databases that are: Yale face database, AT&T (formerly Olivetti Research Laboratory) face database, and METU Vision Laboratory face database. Experiment results are compared with respect to the effects of changes in illumination, pose and expression.

Kernel Eigenface and Fisher LDA show slightly better performance with respect to Eigenfaces method under changes in illumination. Expression differences did not affect the performance of Eigenfaces method.

From test results, it can be observed that Eigenfaces approach is an adequate method that can be used in face recognition systems due to its simplicity, speed and learning capability. By this way, it can easily be used in real time systems.

Keywords: Face recognition, eigenface, principal component analysis, subspace LDA, Kernel eigenface.

ÖZ

ALT-UZAY TABANLI YÜZ TANIMA YÖNTEMLERİNİN KARŞILAŞTIRILMASI

Gönder, Özkan
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Uğur Halıcı

Ağustos 2004, 95 sayfa

Bu tezde yüz tanımada kullanılan farklı yöntemler üzerinde çalışılmıştır. Bu yöntemler Ana Bileşen Analizi(Özyüz), Kernel Özyüz ve Fisher Doğrusal Ayırtaç Analizleridir (DAA). Ana Bileşen Analizi yüzü en iyi tanımlayabilen hesaplayıcı model oluşturabilmek için yüz resmi içindeki en önemli bilgileri çıkarmaya çalışır. Özyüz yönteminde yüzler arasındaki farklılıklar eğitim kümesinde yer alan yüzlerin kovaryans matrisinin özvektörlerinin(özyüzlerin) hesaplanmasıyla bulunur. Özvektörler (Özyüzler) elde edilerek diğer yüzler bu özvektörlerın lineer kombinasyonu ile ifade edilir Tanıma olayı yeni yüzün elde edilen bu özyüzler tarafından oluşturulan yüz uzayına projeksiyonu ile ve projeksiyon sonucu elde edilen konumlar sisteme tanıtılan yüzlerin konumlarıyla karşılaştırılnasıyla yapılmaktadır. Kernel Özyüz yönteminde, Ana Bileşen Analizinden önce eğitim seti lineer olmayan üst uzaya projeksiyonu

yapılmaktadır. Böylelikle yüzlerin lineer olarak değişmeyen özelliklerinin(arka planda farklılıkları, ışık değişiklikleri ve yüz ifadeleri vb.) de ele alınması hedeflenmiştir. Fisher Doğrusal Ayırtaç Analizinde (DAA), Doğrusal ayırtaç analizi Ana Bileşen Analizinden sonra sınıflar arasında ayrımı arttırmak için uygulanmaktadır.

Bu yöntemler üç veritabanı üzerinde uygulanmıştır. Bunlar: Yale veritabanı, AT&T veritabanı ve ODTÜ Görüntü İşleme Laboratuvarı yüz veritabanıdır. Deney sonuçları ışık, poz ve ifade farklılıklarına göre değerlendirilmiştir.

Kernel Özyüz ve Fisher Doğrusal Ayırtaç Analizi Özyüz yöntemine göre ışık farklılıklarında daha iyi performans göstermektedirler. Yüz ifade değişikliği Özyüz yönteminin performansını etkilememektedir.

Test sonuçları Özyüz yönteminin kolay olması, hızlı ve özel donanım kullanmadan bile gerçek zamanda çalışabilmesi sebebiyle yüz tanıma sistemlerinde kullanılabilir bir yöntem olduğunu göstermektedir.

Anahtar Kelimeler : Yüz tanıma, özyüz, ana bileşen analizi, altuzay doğrusal ayrışım analizi, Kernel özyüz.

To My Family!

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. Uğur HALICI for her guidance, advice, criticism and encouragements throughout my research. Special thanks to my family; my father Ömer GÖNDER, my mother Emine GÖNDER and my brother Özer GÖNDER for their great love and support on me to complete this study. I would also like to thank to Murat ÖNDER for his supports, useful discussions, and great friendship throughout my thesis study.

This thesis is partially supported under project BAP-2002-07-04-04, 2002-2004 Face Recognition and Modeling.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

CHAPTER 1

INTRODUCTION

Face recognition has an important part in human activities. The way we interact with other people is firmly based on our ability to recognize them. Face recognition from still and video images is an active research area with numerous commercial and law enforcement applications. Face recognition systems can be used to allow access to an ATM machine o a computer, to control the entry of people into restricted areas, to recognize people in specific areas (banks, stores), or in a specific database (police database). Biometrics are methods to automatically verify or identify individuals using their physiological or behavioral characteristics.

Biometric technologies include

- Face Recognition
- FingerPrint Identification
- Hand Geometry Identification
- Iris Identification
- Voice Recognition
- Signature Recognition
- Retina Identification

A face recognition system would allow user to be identified by a surveillance camera. Human beings often recognize one another by unique

facial characteristics. One of the newest biometric technologies, automatic face recognition, is based on this feature. Face recognition is the most successful form of human surveillance. Face recognition technology is being used to improve human efficiency when recognizing faces, is one of the fastest growing fields in the biometric industry.

The purpose of this thesis is to develop a face recognition system that operates under various conditions such as varying illuminations and backgrounds.

Subspace based methods for face recognition are studied in this thesis. These methods are PCA (Principal Component Analysis), Kernel Eigenface and Fisher LDA.

These methods are implemented on three databases that are: Yale face database, AT&T (formerly Olivetti Research Laboratory) face database, and METU Vision face database. Experiment results are compared with respect to the effects of changes in illumination, pose and expression.

1.1 Thesis Outline

In Chapter 2, the overview of face recognition methods in literature are introduced.

In Chapter 3, the statistical pattern recognition methods used in the thesis are presented. Taking these into account, the selected methods -Principal Component Analysis (Eigenfaces), Kernel Eigenfaces, Linear Discriminant Analysis (LDA) and the motivation around these topics are explained in details

In Chapter 4, applications of the selected methods for face recognition are discussed. Simulation results of these methods with the common face databases: Yale face database, AT&T (formerly ORL) face database and Computer Vision and METU Vision face database are compared.

In Chapter 5, concluding comments and some future directions are given.

"METU Vision" is used for "Intelligent Systems Research Laboratory in Electrical and Electronics Engineering Department of METU" in this thesis.

CHAPTER 2

A SURVEY ON

FACE RECOGNITION METHODS

Face recognition is a pattern recognition task that classifies a face either "known" or "unknown", after comparing it with previously known individuals. Some of the algorithms that have been developed for face recognition are overviewed in this chapter. These algorithms may be broadly classified as template-matching methods, feature-based methods, Karhunen-Loeve Expansion-Based Methods, Linear Discriminate-Based Methods or more recently, model-based methods. For each of these approaches some of the most popular algorithms will be briefly described. These algorithms are summarized and compared in terms of their accuracy, robustness and complexity.

2.1 Template matching

This technique consists of representing an image as single or multiple arrays of pixel values. The arrays are compared with single or multiple templates representing the faces in the training set via a suitable metric. The features of interest can be located manually or by using a more sophisticated automatic approach based on a multi-layer perceptron as detailed in Hutchinson and Welsh [1], a deformable template as described by Yuille [4] or an active contour model

(snake) as reported by Huang and Chen [37] and as originally described by Kass [5].

2.1.1 Isodensity line maps

A different template-based approach was proposed by Nakamura [8]. The technique they presented making use of grey-level isodensity line maps to represent face images. Summarized in their own words, if the brightness of an image is viewed as the height of a mountain, then an isodensity line corresponds to contour lines of equal altitude. A database of 10 subjects with one training image and one test image for each subject was used. Three subjects wore glasses, two men had a thin beard and two women had different make-up and hair styles in the test and training images. Recognition experiments were carried out and successful results were reported.

2.1.2 Multiple template correlation methods

One of the first studies based on multiple template representation was carried out by Baron [6]. A database of 42 subjects was used and each was represented by up to five manually selected face features (full face, mouth, right eye, chin and hair), and each face feature contained up to four distinct templates. A total of up to 20 pictorial templates were stored for each subject, with each template being a 15x16 array of pixels. A test image was first reduced to a 15x16 full face array and then compared with each full face template in the training set. If the correlation value between the reduced test image and one of the full face

templates exceeded a threshold of recognition, the test image was recognized as the corresponding subject. If the correlation value fell between the threshold of recognition and a lower value called the threshold of recall, then the other face features were recalled and used for recognition. If for at least three out of four of the features the correlation value exceeded the threshold of recall, the test image was recognized as the current subject.

More recently Brunelli and Poggio [7] presented results based on a similar approach. They used a database of 47 subjects, where each subject was represented by a full frontal image and a set of four templates (eyes, nose, mouth and the whole face). Recognition of a test image was performed by computing a normalized cross correlation for each template and by finding the highest cumulative score.

2.1.3 Vector quantised templates

Sutherland [9] used a template-based approach, where each of the original eight feature templates they selected was substituted with an approximately similar template drawn from a code-book via vector quantisation. Various algorithms can be used to generate useful code-books and two such algorithms were presented by Ramsay [10]. Using a database of 30 subjects with 10 training images and 10 test images for each subject, a successful recognition rate of 89% was reported.

2.1.4 Neural network based template matching

Templates have been used as input to neural network based systems. Allinson [11] used a 32x32 full image template and two 64x32 templates for the eye and mouth regions respectively. These templates were used as inputs to Kohonen's [12] self-organising feature maps. The maps produced a topology which preserved the structure of the input templates. The maps were used as input to a multi-layer perceptron which carried out the classification. Other work by Cottrell and Fleming [13] studied the performance of a network that automatically extracted features (the output of the hidden units) from a 64x64 full face template and input them to a one-layer network for identity and also gender classfication. Test images were perfectly identified with a database of 11 subjects. A gender recognition success performance of 37% was reported. Stonham [14] detailed experiments on face recognition using a general purpose pattern recognition machine called WISARD. A database of 16 subjects was used and full image 153x214 templates were input to a self-adapting single layer network. Subjects were asked to appear before a camera, face on, for approximately 20 seconds. On average, 200-400 images were required to complete the training. Real time testing results were reported with error free recognition rates.

2.2 Feature Based Face Recognition

It was mentioned before that, there were two basic approaches to the face recognition problem: Feature based face recognition and principal component analysis methods. Although feature based face recognition can be divided into two different categories, based on frontal views and profile silhouettes, they share some common properties and we will treat them as a whole. In this section, basic principals of feature based face recognition from frontal views are presented.

The first step of human face identification is to extract the features from facial images. In the area of feature selection, the question has been addressed in studies of cue salience in which discrete features such as the eyes, mouth, chin and nose have been found important cues for discrimination and recognition of faces.

After knowing what the effective features are for face recognition, some methods should be utilized to get contours of eyes, eyebrows, mouth, nose, and face. For different facial contours, different models should be used to extract them from the original portrait. Because the shapes of eyes and mouth are similar to some geometric figures, they can be extracted in terms of the deformable template model [15]. The other facial features such as eyebrows, nose and face are so variable that they have to be extracted by the active contour model [16, 17]. These two models can be illustrated in the following:

- **Deformable template model**

The deformable templates are specified by a set of parameters which uses a priori knowledge about the expected shape of the features to guide the contour deformation process. The templates are flexible enough to change their size and other parameter values, so as to match themselves to the data. The final values of these parameters can be used to describe the features. This method works well regardless of variations in scale, tilt, and rotations of the head. Variations of the parameters should allow the template to fit any normal instance of the feature. The deformable templates interact with the image in a dynamic manner. An energy function is defined which contains terms attracting the template to salient features such as peaks and valleys in the image intensity, edges and intensity itself. The minima of the energy function correspond to the best fit with the image. The parameters of the template are then updated by steepest descent.

- **Active contour model (Snake)**

The active contour or snake is an energy minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. Snakes lock onto nearby edges, localizing them accurately. Because the snake is an energy minimizing spline, energy functions whose local minima comprise the set of alternative solutions to higher level processes should be designed. Selection of an answer from this set is accomplished by the addition of energy terms that push the model toward the

desired solution. The result is an active model that falls into the desired solution when placed near it. In the active contour model issues such as the connectivity of the contours and the presence of corners affect the energy function and hence the detailed structure of the locally optimal contour. These issues can be resolved by very high-level computations.

2.2.1 Effective Feature Selection

Before mentioning the facial feature extraction procedures, we have the following two considerations:

- The picture-taking environment must be fixed in order to get a good snapshot.
- Effective features that can be used to identify a face efficiently should be known.

Despite the marked similarity of faces as spatial patterns we are able to differentiate and remember a potentially unlimited number of faces. With sufficient familiarity, the faces of any two persons can be discriminated. The skill depends on the ability to extract invariant structural information from the transient situation of a face, such as changing hairstyles, emotional expression, and facial motion effect. Features are the basic elements for object recognition. Therefore, to identify a face, we need to know what features are used effectively in the face recognition process. Because the variance of each feature associated

with the face recognition process is relatively large, the features are classified into three major types:

- **First-order features values.** Discrete features such as eyes, eyebrows, mouth, chin, and nose, which have been found to be important [11] in face identification and are specified without reference to other facial features, are called first-order features. Important first-order features are given in Table 2.1.

Table 2.1 First-order features

Measurement	Facial Location
Area, angle	Left eyebrow Right eyebrow Left eye Right eye mouth face
Distance	Length of left eyebrow Length of right eyebrow Length of left eye Length of right eye Length of mouth Length of face Height of face

- **Second-order features values.** Another configural set of features which characterize the spatial relationships between the positions of the first-order features and information about the shape of the face are called second-order features. Important second-order features are given in Table 2.2. Second order features that are related to nose, if nose is noticeable are given in Table 2.3.

Table 2.2 Second-order features

Measurement	Facial Location
Distance	Left eyebrow <-> right eyebrow Left eye <-> right eye Left eyebrow <-> left eye Right eyebrow <-> right eye Left eyebrow <-> mouth Right eyebrow <-> mouth Left eye <-> mouth Right eye <-> mouth Eyebrow <-> side of face Eye <-> side of face Mouth <-> side of face Mouth <-> lower part of face
Angle	Left eyebrow – left eye – left eyebrow Right eyebrow – right eye – right eyebrow Left eye – left eyebrow – left eye Right eye – right eyebrow – right eye Left eyebrow - mouth – right eyebrow Left eye - mouth – right eye Left eyebrow - mouth – right eye Left eyebrow – left eye – mouth Right eyebrow - right eye - mouth

Table 2.3 Features related to nose, if nose is noticeable

Measurement	Facial Location
Distance	Left nose <-> right nose Left eyebrow <-> left nose Right eyebrow <-> right nose Left eye <-> left nose Right eye <-> right nose Left nose <-> mouth Right nose <-> mouth
Angle	Left eyebrow – center of nose – right eyebrow Left eye – center of nose – right eye Left nose – mouth - right nose Left eyebrow - left eye - left nose Right eyebrow - right eye - right nose

- **Higher-order feature values.** There are also higher-order features whose values depend on a complex set of feature values. For instance, age might be a function of hair coverage, hair color, skin tension, presence of wrinkles and age spots, forehead height which changes because of receding hairline, and so on.

Variability such as emotional expression or skin tension exists in the higher- order features and the complexity, which is the function of first-order and second-order features, is very difficult to predict. Permanent information belonging to the higher-order features can not be found simply by using first and second-order features. For a robust face recognition system, features that are invariant to the changes of the picture taking environment should be used. Thus, these features may contain merely first-order and second-order ones. These effective feature values cover almost all the obtainable information from the portrait. They are sufficient for the face recognition process.

The feature values of the second-order are more important than those of the first-order and they are dominant in the feature vector. Before mentioning the facial feature extraction process, it is necessary to deal with two preprocessing steps:

- **Threshold assignment.** Brightness threshold should be known in order to discriminate the feature and other areas of the face. Generally, different

thresholds are used for eyebrows, eyes, mouth, nose, and face according to the brightness of the picture.

- **Rough Contour Estimation Routine (RCER).** The left eyebrow is the first feature that is to be extracted. The first step is to estimate the rough contour of the left eyebrow and find the contour points. Generally, the position of the left eyebrow is about one-fourth of the facial width. Having this a priori information, the coarse position of the left eyebrow can be found and its rough contour can be captured. Once the rough contour of the left eyebrow is established, the rough contours of other facial features such as left eye, right eyebrow, mouth or nose can be estimated by RCER [20]. After the rough contour is obtained, its precise contour will be extracted by the deformable template model or the active contour model.

2.2.2 Feature Extraction Using the Deformable Templates

After the rough contour is obtained, the next step of face recognition is to find the physical contour of each feature. Conventional edge detectors can not find facial features such as the contours of the eye or mouth accurately from local evidence of edges, because they can not organize local information into a sensible global perception. There is a method to detect the contour of the eye by the deformable template which was originally proposed by Yullie [4]. It is possible to

reduce the amount of computations at the cost of the precision of the extracted contour.

2.2.2.1 *Eye Template*

The deformable template acts on three representations of the image, as well as on the image itself. The first two representations are the peak and valleys in the image intensity and the third is the place where the image intensity changes quickly. The eye template developed by Yullie et al. consists of the following features:

- A circle of radius r, centered on a point (x_c, y_c), corresponding to the iris. The boundaries of the iris and the whites of the eyes are attracted to edges in the image intensity. The interior of the circle is attracted to valleys, or low values in the image intensity.
- A bounding contour of the eye attracted to edges. This contour is modeled by two parabolic sections representing the upper and lower parts of the boundary. It has a center (x_e, y_e), with 2w, maximum height h1 of the boundary above the center, maximum height h2 of the boundary below the center, and an angle of rotation Φ.
- Two points, corresponding to the centers for the whites of the eyes, which are attracted to peaks in the image intensity.
- Regions between the bounding contour and the iris which also correspond to the whites of the eyes. These will be attracted to large intensity values.

The original eye template can be modified for the sake of simplicity where the accuracy of the extracted contour is not critical. The lack of a circle does not affect the classified results because the feature values are obtained from other information. The upper and lower parabola will be satisfactory for the recognition process. Thus, the total energy function for the eye template can be defined as a combination of the energy functions of edge, white and black points.

The total energy function is defined as

$$E_{total} = E_{edge} + E_{white} + E_{black} \tag{2.1}$$

where E_{edge}, E_{white} and E_{black} are defined in the following:

- The edge potentials are given by the integral over the curves of the upper and lower parabola divided by their length:

$$E_{edge} = -\frac{w_1}{upper-length}\int_{upper-bound}\Phi_{edge}(x,y)ds - \frac{w_2}{lower-length}\int_{lower-bound}\Phi_{edge}(x,y)ds \tag{2.2}$$

where upper-bound and lower-bound represent the upper and lower parts of the eye, and Φ_{edge} represents the edge response of the point (x,y).

- The potentials of white and black points are defined as the integral over the area bounded by the upper and lower parabola divided by the area:

$$E_{w,b} = -\frac{1}{Area}\iint_{para-area}(-w_b N_{black}(x,y) + w_w N_{white}(x,y))dA \tag{2.3}$$

where $N_{black}(x, y)$ and $N_{white}(x, y)$ represent the number of black and white points, and w_b, w_w are weights related with black and white points.

In order to be not affected by an improper threshold, the black and white points in Eq. (2.3) are defined as:

- P(x,y) is a black point if I(x,y) $\leq$ (threshold - tolerance),
- P(x,y) is a white point if I(x,y) $\geq$ (threshold + tolerance),
- P(x,y) is an unambiguous point if I(x,y) is in-between. (2.4)

where I(x,y) is the image intensity at point (x,y).

By the energy functions defined above, we can calculate the energy in the range of little modulations of 2w, h_1 , h_2 and Φ. When the minimum energy value takes place, the precise contour is extracted.

2.2.2.2 *Mouth Template*

In the whole features of the front view of the face, the role of the mouth is relatively important. The properties of the mouth contour are heavily involved in the face recognition process. The deformable mouth template changes its own shape when it comes across the image areas of edge (which the intensity changes quickly), and white and black points. Generally, features related to middle lips, lower and upper lips are extracted. Because of the effect of brightness in the picture taking period, the middle of the lower lip may not be apparent. RCER can not find the approximate height of the lower lip. Fortunately, the length of the

mouth can still be found by RCER. Usually, the height of the lower lip is between one-fourth and one-sixth of the mouth's length. The mouth contour energy function consists of the edge term E edge and the black term E black . The edge term dominates at the edge area, where as the black term encloses as many black points belonging to the mouth as possible.

$$E_{total} = E_{edge} + E_{black} \tag{2.5}$$

The edge energy function consists of three parts: middle lip (gap between lips), lower lip and upper lip separated at philtrum. The equation of the middle lip part is

$$E_{edge} = -\frac{w_{lower}}{lower - length}\int_{lower}\Phi_{edge}(x,y)ds - \frac{w_{left}}{left - length}\int_{left}\Phi_{edge}(x,y)ds - \frac{w_{right}}{right - length}\int_{right}\Phi_{edge}(x,y)ds \tag{2.6}$$

where lower represents the lower boundary of mouth, *left* represents the left part of upper lip, *right* represents the right part of upper lip, and $\Phi_{edge}(x,y)$ represents the edge response of point (x,y).

The black energy function helps the edge energy to enclose black points belong to the mouth and are defined as:

$$E_{black} = \frac{1}{Area}\int_{Lbound}^{Ubound}\int -W_{black}N_{black}(x,y)dA + \frac{1}{mid\_length}\int_{mid} -w_{mid}N_{black}(x,y)dS \tag{2.7}$$

where L_{bound} represents lower lip, U_{bound} represents upper lip, and *mid* represents number of black points. The black points are defined by Eq. (2.4). The weights w_{black} , w_{mid} , w_{lower} , w_{left} and w_{right} are experimentally determined.

2.3 Karhunen-Loeve Expansion-Based Methods

2.3.1 The Eigenfaces Method

One of the implemented methods in this study is the Eigenfaces method, which is briefly explained here. Detailed theoretical information is given in **Section 3** and the test results for this method can be seen in **Section 4.**

The Eigenfaces method proposed by Turk and Pentland is based on the Karhunen-Loeve Transform (KLT), and is motivated by the earlier work of Sirovitch and Kirby for efficiently representing face images. The eigenvectors of the covariance matrix C of the ensemble of training faces are called eigenfaces. The space spanned by the eigenvectors v_k, k = 1.. K corresponding to the K largest eigenvalues of the covariance matrix, is called the face space. A new face image is transformed into its eigenface components by projection onto the face space. The projections form the feature vector which describes the contribution of each eigenface in representing the input image. A test image is recognized by computing the Euclidian distance in the feature space and selecting the closest match. The effect of lighting conditions over the KLT method has been detailed in [21]. The eigenface method has also been used for face detection by measuring

the distance from each local pattern in a test image to the face space defined by the eigenfaces.

2.3.2 The "Parametric" Approach versus the "View-Based" Approach

In [22] Murase and Nyar extended the capabilities of the eigenface method to general 3D object recognition under different illumination and viewing conditions. Given N object images taken under P views and L different illumination conditions, a set of eigenvectors was obtained by applying the eigenface method to all the available data. In this way a single "parametric space" describes the object identity as well as the viewing or illumination conditions. The eigenface decomposition of this space was used for feature extraction and classification. However, in order to ensure discrimination between different objects, the number of eigenvectors used in this method was increased compared to the classical Eigenface method.

Pentland et. Al. [3] developed a "view-based" eigenface approach for human face recognition under general viewing conditions. The "view-based" approach is essentially an extension of the eigenface technique to multiple sets of Eigenvectors, one for each face orientation. First, the orientation of the test face is determined by calculating the residual description error(distance from feature space) for each view space, and selecting the space for which the distance is minimized. Once the proper view is determined, the face image is classified using the eigenface method in the corresponding space. As expected, the view based

representation has better recognition results than the parametric approach, at a cost of higher computational complexity.

2.3.3 Recognition Using Eigenfeatures

While the classical eigenface method uses the KLT coefficients of the template corresponding to the whole face image, in [3] Pentland et. Al. introduced a face detection and recognition system that uses the KLT coefficients of the templates corresponding to the significant facial features such as eyes, nose and mouth. For each of the facial features, a feature space is built by selecting the most significant "eigenfeatures", which are the eigenvectors corresponding to the largest eigenvalues of the features correlation matrix. The significant facial features were detected using the distance from the feature space and selecting the closest match. The scores of similarity between the templates of the test image and the templates of the images in the training set were integrated in a cumulative score that measures the distance between the test image and the training images. The method was extended to the detection of features under different viewing geometries by using either a view-based eigenspace or a parametric eigenspace.

2.4 Linear Discrimination-Based Methods

Like Eigenface Method one of the implemented methods in this study is the Fisher Linear Discriminant Analysis method, which shall be briefly explained in this section. Detailed theoretical information shall be given in **Section 3** and the test results for this method can be seen in **Section 4.**

In [29], the authors proposed a new method for face recognition using Fisher's Linear Discriminant Transform (LDT). The "Fisherface" method uses the class membership information and develops a set of feature vectors in which variations of different faces are emphasized while different instances of faces due to illumination conditions, facial expressions, and orientations, are de-emphesized. While the Karhunen_Loeve Transform performs a rotation on a set of axes along which the projection of sample vectors differ most in the autocorrelation sense, the LDT performs a rotation on a set of axes along which the projection of sample vectors show maximum discrimination. Each test image is projected onto the optimal LDT space and the resulting set of coefficients is used to compute the Euclidean distance from the images in the training set. More recently the Fisherface method has also been applied to face detection from color images [30].

In [27], Akamatsu et. al. applied LDT to the magnitude of the Fourier Spectrum of the intensity image. The database used in the experiments contained large variations in lighting conditions as well as variations in head orientation. The results reported by the authors showed that LDT in the Fourier domain is significantly more robust to variations in lighting than the LDT applied directly to the intensity images. However, the computational complexity of this method is significantly higher than the classical Fisherface method due to the computation of the Fourier spectrum.

2.5 Model-Based Methods

2.5.1 Hidden-Markov Model Based Methods

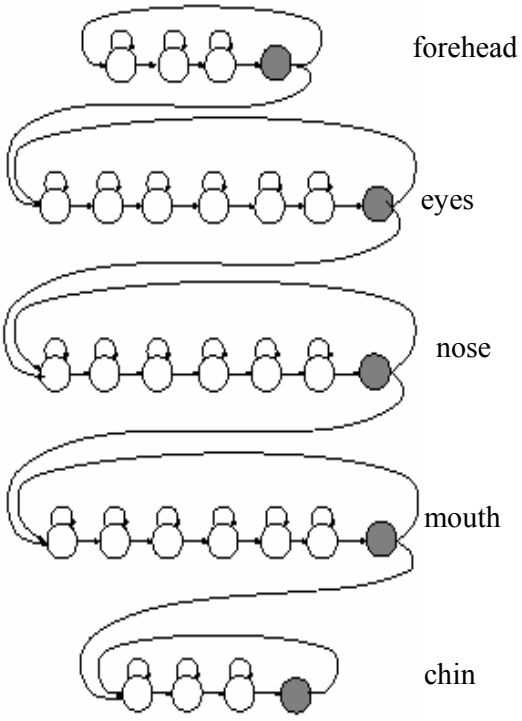


Figure 2.1: Hidden Markov Model Face Recognition

Hidden Markov Models (HMM) are a set of statistical models used to characterize the statistical properties of a signal [23], [24]. HMM's have been used extensively for speech recognition, where data is naturally one-dimensional (1D) along the time axis. However, the equivalent fully-connected two-dimensional HMMwould lead to a very high computational problem [25].

Attempts have been made to use multi-model represenatations that lead to pseudo-two-dimensional HMMs [26].

In [28], Samaria et. al. Proposed using the 1D continuous HMM for face recognition. Assuming that each face is an upright, frontal position, featured will occur in a predictable order, i.e. forehead, eyes, nose etc. This ordering suggests the use of a top-to-bottom model, where only transitions between adjacent states in a top to bottom manner are allowed [31]. The states of the model correspond to the significant facial features such as forehead, eyes, nose, mouth, and chin [32]. The observation sequence O is genereated from an X x Y image using an X x L sampling window with X x M pixels overlap. Each observation vector is a block of L lines. There is an M line overlap between successive observations [33].

Given c face images for each subject of the training set, the goal of the training stage is to optimze the parameters of the HMM to best describe the observations in the sense of maximizing the probability of the observations given the model. Recognition is carried out by matching the test image against each of the trained models. To do this, the image is converted to an observation sequence.

In [34], Samaria increased the number of states used to characterize each of the significant facial features. The observation sequence used with this model is obtained by sliding a rectangular window from left to right and from top to bottom of the image and using the pixels intensities extracted from each window as observation vectors. To preserve the two- dimensional structure of the data, a

marker block was added at the end of each line in the image, and an additional end-of-line state was aded to the structure of each horizontal HMM (Figure 2.1-a). The end-of-line states are allowed two transitions: one to the same row of states, and one to the next row of states. It was found [34] that by stating the initial standard deviation of the End-of-line states to be small, and the means close to the intensity of the-end-of-line marker block, the state topology was preserved and the parameters of the end-of-line states were unaltered after re-estimation. In the same work, it was shown that similar recognition results were obtained for the unconstrained P2D-HMM structure (Figure 2.1-b). However, this topology allows transition to a state corresponding to another facial feature from a block that is not at the end of a row, and consequently does not preserve the two dimensional structure of the data. Preliminary results showed that for this structure, the face recognition results are as high as 95%. However, due to the large dimension of the observation vectors used, the system required about four minutes for a face to be recognized on Sparc 20 workstations.

2.6 Summary of the Related Work

Some of the most successful approaches to face recognition were analyzed. Due to the fact that these methods were tested on different databases, a quantitative comparison is not meaningful. The recognition results of the approaches discussed are summarized in Table 2.4. In general, template-based methods performed at high accuracy when the size of the images is fixed. In this

study, Eigenface approach, a modified version of Eigenfaces: Kernel Eigenfaces and Fisherface Methods are implemented.

All selected methods in this thesis are investigated in different cases and their performances are compared against these conditions. Three face image databases are chosen for these tests. These are: Yale Database, AT&T (Formerly ORL Face Database) and METU Vision Lab Face Database. Each Database have different properties in order to measure the performances of the selected methods in different conditions.

In the following sections, first, thoretical details of the implemented methods are explained and then test results of these algorithms and comments on these test results are presented.

Table 2.4 Comparison of some of the face recognition approaches

Approach	Training Set	Training Set	Recognition Rate	Timing
Correlation [2]	47 subjects	47 subjects 4 images per subject	100%	Not specified
Correlation [18]	62 subjects 15 images per subject	62 subjects 10 images per subject	98.7%	10-15 min on Sparc 2
Eigenface [19]	16 subjects one image per subject	2500 images from 16 subjects	variations in: size 64% orientation 85% lighting 96%	350 msec Sparc 1
Eigenface Parametric [3]	21 subjects	21 subjects 9 images per subject	78-88%	higher than the view-based approach
Eigenface View based [3]	21 subjects	21 subjects 9 images per subject	83-90%	lower than the parametric-based approach
Eigenfeatures [3]	45 subjects one image per subject	45 subjects one image per subject	95-98%	Not specified
KL-FSAT [27]	269 subjects one image per subject	100 images 5 images per subject	91%	higher than eigenface method
Fisherfaces [29]	16 subjects 9 images per subject	16 subjects one image per subject	99.4%	lower than eigenfaces

Table 2.4 Comparison of some of the face recognition approaches (cont'd)

Approach	Training Set	Training Set	Recognition Rate	Timing
SVD [35], [37]	8 subjects 3 images per subject	40 images 5 images per subject	100%	high due to SVD calculation
Auto Association and Classification NN [42], [50]	40 subjects 5 images per subject	40 subjects 5 images per subject	20%	Not specified
PDBNN [45]	40 subjects 5 images per subject	40 subjects 5 images per subject	96%	0.1 sec on SGI Indy 100 MHz
Convolutional NN [43]	40 subjects 5 images per subject	40 subjects 5 images per subject	96.2%	0.5 sec on SGI Indy 100 MHz
Dynamic Link Matching [48], [50]	40 subjects 5 images per subject	40 subjects 5 images per subject	80%	Not specified
VFR [49]	40 subjects 5 images per subject	40 subjects 5 images per subject	92.5%	320 sec on Pentium 200MHz
HMM [36]	40 subjects 5 images per subject	40 subjects 5 images per subject	85%	12 sec on Sparc 2
HMM [34]	40 subjects 5 images per subject	40 subjects 5 images per subject	90-95%	4 min on Sparc 2

CHAPTER 3

IMPLEMENTED FACE RECOGNITION METHODS

In this chapter, the theoretical background of the implemented methods are explained in detail. These methods are:

- Eigenface Method (PCA),
- Kernel Eigenfaces,
- Subspace LDA (PCA + LDA).

3.1 Face Recognition by Eigenfaces

The Eigenface method tries to find a lower dimensional space for the representation of the face images by eliminating the variance due to non-face images; that is, it tries to focus on the variation just coming out of the variation between the face images.

Eigenface method is the implementation of Principal Component Analysis (PCA) over face images. In this method, the features of the studied images are obtained by looking for the maximum deviation of each image from the mean image. This variance is obtained by getting the eigenvectors of the covariance matrix of all the images.

The eigenface space is obtained by applying the eigenface method to the training images. Then, the training images are projected into the eigenface space.

Next, the test image is projected into this new space and the distance of the projected test image to the training images is used to classify the test image. In the standard eigenface procedure proposed by Turk and Pentland [19], Nearest Mean Classifier is used for the classification of test images.

In mathematical terms Image I is:

$$(N_x \text{ x } N_y) \text{ pixels} \tag{3.1.1}$$

The image matrix I of size (N_x x N_y) pixels is converted to the image vector Γ of size (P x 1) where P = (N_x x N_y); that is the image matrix is reconstructed by adding each column one after the other. Training set Γ is set of image vectors and its size is (P x M_t) where M_t is the number of the training images:

$$\Gamma = [\Gamma_1 \Gamma_2 ... \Gamma_{Mt}] \tag{3.1.2}$$

Mean face Ψ is the arithmetic average of the training image vectors at each pixel point and its size is (P x 1):

$$\Psi = \frac{1}{M_t} \sum_{i=1}^{M_t} \Gamma_i \tag{3.1.3}$$

Mean subtracted image Φ is the difference of the training image from the mean image (size P x 1):

$$\Phi = \Gamma - \Psi \tag{3.1.4}$$

Difference Matrix A is the matrix of all the mean subtracted training image vectors and its size is (P x M_t):

$$A = [\Phi_1 \Phi_2 \Phi_3 ... \Phi_{Mt}] \tag{3.1.5}$$

Covariance Matrix X is the covariance matrix of the training image vectors of size (P x P):

$$X = A \cdot A^T = \frac{1}{M_t} \sum_{i=1}^{M_t} \Phi_i \Phi_i^T \quad (3.1.6)$$

An important property of the Eigenface method is obtaining the eigenvectors of the covariance matrix. For a face image of size (N_x x N_y) pixels, the covariance matrix is of size (P x P), P being (N_x x N_y). This covariance matrix is very hard to work with due to its huge dimension causing computational complexity. On the other hand, Eigenface method calculates the eigenvectors of the (M_t x M_t) matrix, M_t being the number of face images, and obtains (P x P) matrix using the eigenvectors of the (M_t x M_t) matrix.

Initially, a matrix Y which is of size (M_t x M_t) is defined as:.

$$Y = A^T \cdot A = \frac{1}{M_t} \sum_{i=1}^{M_t} \Gamma_i^T \Gamma_i \quad (3.1.7)$$

Then, the eigenvectors v_i and the eigenvalues μ_i of Y are obtained,

$$Y.v_i = \mu_i \cdot v_i \quad (3.1.8)$$

The value of Y is put in this equation,

$$A^T \cdot A \cdot v_i = \mu_i \cdot v_i \quad (3.1.9)$$

Both sides are left multiplied by A:

$$A \cdot A^T \cdot A \cdot v_i = A \cdot \mu_i \cdot v_i \quad (3.1.10)$$

The necessary matrix arrangements are performed,

$$A \cdot A^T \cdot A \cdot v_i = \mu_i \cdot A \cdot v_i \quad (3.1.11)$$

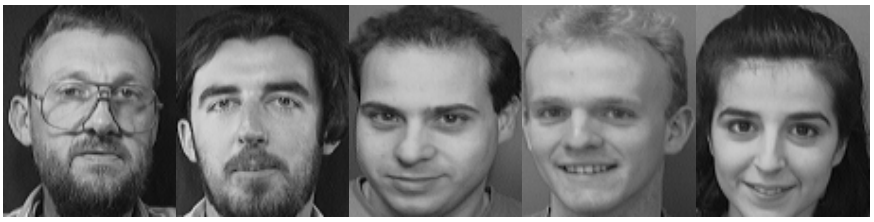
(as μ_i is a scalar, this arrangement can be done)

$$X \cdot A \cdot v_i = \mu_i \cdot A \cdot v_i \tag{3.1.12}$$

Now group $A \cdot \mu_i$ and call a variable $v_i = A \cdot \mu_i$. It is easy to see that

$$v_i = A \cdot \mu_i \tag{3.1.13}$$

is one of the eigenvectors of X = $A \cdot A^T$ and its size is (P x 1). Thus, it is possible to obtain the eigenvectors of X by using the eigenvectors of Y. A matrix of size (M_t x M_t) is utilized instead of a matrix of size (P x P) (i.e. [{N_x x N_y} x {N_x x N_y}]). This formulation brings substantial computational efficiency. In Figure 3.1, some example images and mean image of the images from the AT&T database are given. Some characteristic eigenfaces obtained from this database can be seen. The eigenfaces are in fact (P x 1) vectors for the computations; in order to see what they look like, they are rearranged as (N_x x N_y) matrices.



(a)

(b)

Figure 3.1 (a) Example images from AT&T database
(b) Mean face obtained from the AT&T database

Instead of using M_t of the eigenfaces, $M' \leq M_t$ of the eigenfaces can be used for the eigenface projection. This is achieved to eliminate some of the eigenvectors with small eigenvalues, which contribute less variance in the data.

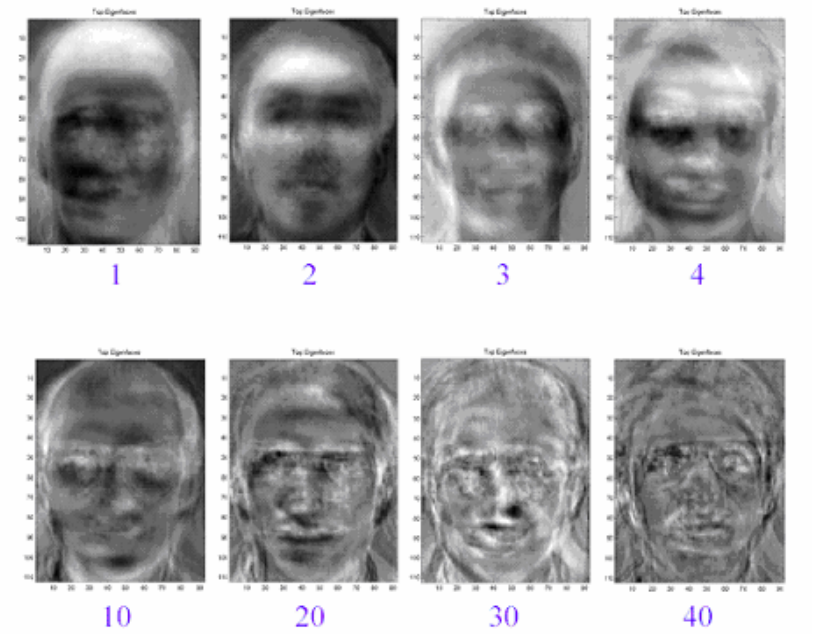


Figure 3.2: Some example of eigenfaces sorted with respect to their eigenvalues

Most of the generalization power is contained in the first few eigenvectors. For example, 40% of the total eigenvectors have 85 – 90% of the total generalization power. Thus, using 40% of the total number of eigenvectors may end up with reasonable classification results.

Eigenvectors can be considered as the vectors pointing in the direction of the maximum variance and the value of the variance the eigenvector represents is directly proportional to the value of the eigenvalue (i.e. the larger the eigenvalue indicates the larger variance the eigenvector represents). Hence, the eigenvectors are sorted with respect to their corresponding eigenvalues. The eigenvector having the largest eigenvalue is marked as the first eigenvector, and so on. In this manner, the most generalizing eigenvector comes first in the eigenvector matrix.

In the next step, the training images are projected into the eigenface space and thus the weight of each eigenvector to represent the image in the eigenface space is calculated. This weight is simply the dot product of each image with each of the eigenvectors. Projection w_k is the representation of the training image in the eigenface space and its size is (M' x 1):

$$w_k = v_k^T \cdot \Phi = v_k^T \cdot (\Gamma - \Psi) \tag{3.1.14}$$

Weight Matrix (M' x 1) Ω is:

$$\Omega = \left[w_1 \;\; w_2 \;\; \ldots \; w_{M'} \right]^T \tag{3.1.15}$$

At this point, the images are just composed of weights in the eigenface space, simply like they have pixel values in the image space. The important aspect of the eigenface transform lies in this property. Each image is represented by an image of size (N_x x N_y) in the image space, whereas the same image is represented by a vector of size (M' x 1) in the eigenface space. Moreover, having the dimension structure related to the variance of the data in hand makes the eigenface representation a generalized representation of the data. This makes the algorithm a solution to the "curse of dimensionality" problem seen in the standard pattern recognition task. When a new test image is to be classified, it is also mean subtracted and projected onto the eigenface space and Nearest Mean algorithm is used for the classification of the test image vector in the standard eigenface method; that is, the test image is assumed to belong to the nearest class by calculating the Euclidean distance of the test image vector to the mean of each class of the training image vectors. Test image vector:

$$\Gamma_T \tag{3.1.16}$$

is the test image vector of size (P x 1).

Mean subtracted image Φ_T is the difference of the test image from the mean image (size P x 1):

$$\Phi_T = (\Gamma_T - \Psi) \tag{3.1.17}$$

Projection w_k is the projection of a training image on each of the eigenvectors where k = 1, 2… M':

$$w_k = v_k^T \cdot \Phi_T = v_k^T \cdot (\Gamma_T - \Psi) \tag{3.1.18}$$

Weight Matrix Ω_T is the representation of the test image in the eigenface space and its size is (M' x 1):

$$\Omega_T = \left[w_1 \;\; w_2 \;\; \ldots \; w_{M'} \right]^T \tag{3.1.15}$$

The training and test image vectors can be reconstructed by a back transformation from the eigenface space to the image vector space. Reconstructed image vector Γ_f is:

$$\Gamma_f = v \cdot \Omega + \Psi = \Phi_f + \Psi \tag{3.1.16}$$

If possible, it is better to work on a database of more than one image per individual in order to increase the robustness to minor changes in expression, illumination and slight variations of view angles. A class of images for an individual can be formed and this class can be considered as the representative image vector of that class. For an individual having q_i images in the database, the

average of the projections of each class is the mean of all the projected image vectors in that class.

In mathematical terms Average Class Projection Ω_Ψ is:

$$\Omega_\Psi = \frac{1}{q_i}\sum_{i=1}^{q_i}\Omega_i \tag{3.1.17}$$

This average class projection can be used as one of the vectors (representing an image class instead of an image vector) to compare with the test image vector. A similarity measure δ_i is defined as the distance between the test image vector and ith face class:

$$\delta_i = \left\|\Omega_T - \Omega_{\Psi_i}\right\| = \sqrt{\sum_{k=1}^{M_T}(\Omega_{Tk} - \Omega_{\Psi ik})^2} \tag{3.1.18}$$

A distance threshold Θ may be defined for the maximum allowable distance from any face class, which is half of the distance between the two most distant classes:

$$\Theta = \frac{1}{2}\max(\left\|\Omega_{\Psi i} - \Omega_{\Psi j}\right\|) \tag{3.1.19}$$

Classification procedure of the Eigenface method ensures that face image vectors should fall close to their reconstructions, whereas non-face image vectors should fall far away. Hence, a distance measure is the distance between the mean subtracted image and the reconstructed image:

$$\varepsilon^2 = \left\| \Phi - \Phi_f \right\|^2 \qquad (3.1.20)$$

The recognition of an image knowing these two measures δ_i and ε can be done as follows:

(a) If $\varepsilon > \Theta \Rightarrow$ the image is not a face(independent of the values of δ_i)

(b) If $\varepsilon < \Theta$ and for all i $\delta_i > \Theta \Rightarrow$ the image is an unknown face,

(c) If $\varepsilon < \Theta$ and for one of i $\delta_i < \Theta \Rightarrow$ the image belongs to the training face class i.

This is the standard Eigenface approach suggested by Turk and Pentland in 1991 and variations of this method is explained in the next sections.

3.2 Face Recognition by Kernel Eigenface Method

Eigenface or Principal Component Analysis (PCA) methods have demonstrated their success in face recognition, detection, and tracking. The representation in PCA is based on the second order statistics of the image set, and does not address higher order statistical dependencies such as the relationships among three or more pixels. Recently Higher Order Statistics (HOS) have been used as a more informative low dimensional representation than PCA for face and vehicle detection. In this chapter a generalization of PCA, Kernel Principal Component Analysis (Kernel PCA), for learning low dimensional representations in the context of face recognition. In contrast to HOS, Kernel PCA computes the higher order statistics without the combinatorial explosion of time and memory

complexity. While PCA aims to find a second order correlation of patterns, Kernel PCA provides a replacement which takes into account higher order correlations. The recognition results using kernel methods with Eigenface methods on two benchmarks are compared. Empirical results show that Kernel PCA outperforms the Eigenface method in face recognition.

Subspace methods have been applied succesfully in aplications such as face recognition using Eigenfaces (or PCA face) [38], object recognition [39], and tracking [40]. Represantations such as PCA encode the pattern information based on second order dependencies, i.e., pixelwise covariance among the pixels, and are insensitive to the dependencies of multiple (more than two) pixels in the patterns. Since the eigenvectors in PCA are the orthonormal bases, the principal components are uncorrelated. In other words, the coefficients for one of the axes cannot be linearly represented from the coefficients of the other axes.

Higher order dependencies in an image include nonlinear relations among the pixel intensity values, such as the relationships among three or more pixels in an edge or a curve, which can capture important information for recognition. Several researchers have conjectured that higher order statistics may be crucial to better represent complex patterns. Recently, Higher Order Statistics (HOS) have been applied to visual learning problems. Use HOS of the images of a target object to get a better approximation of an unknown distribution. Experiments on face detection [41] and vehicle detection [44] show comparable, if no better, results than other PCA-based methods.

$Cum(x_1, x_2) = E[x_1x_2]$

$Cum(x_1, x_2, x_3) = E[x_1x_2x_3]$

$Cum(x_1, x_2, x_3, x_4) = E[x_1x_2x_3x_4] - E[x_1x_2]\mathrm{E}[x_3x_4] - E[x_1x_3]E[x_2x_4] - E[x_1x_4]E[x_2x_3]$

The computation involved in HOS depends on the order of cumulants and is usually heavy because of computing expectations in a high dimensional space.

In contrast to computing cumulants in HOS, a formulation is sought which computes the higher order statistics using only dot products, $\Phi(x_i) \cdot \Phi(x_j)$, of the training patterns x where Φ is a nonlinear projection function. Since we can compute these dot products efficiently, we can solve the original problem without explicitly mapping to R^F. This is achieved using Mercer kernels where a kernel $k(x_ix_j)$ computes the dot product in some feature space R^F, i.e., $k(x_ix_j) = \Phi(x_i) \cdot \Phi(x_j)$.

The idea of using kernel methods has also been adopted in the Support Vector Machines (SVMs) in which kernel functions replace the nonlinear projection functions such that an optimal separating hyperplane can be constructed efficiently [46]. Schölkopf et al. proposed the use of Kernel PCA for object recognition in which the principal components of an object image comprise a feature vector to train a SVM [47]. Empirical results on character recognition using MNIST data set and object recondition using MPI chair database show that Kernel PCA is able to extract nonlinear features. Since much of the important informaton may be contained in the high order relationships

among the pixels of a face image, the use of Kernel PCA for face recognition is investigated and its performance is compared against the Eigenface method and Fisherface method.

Kernel Principal Component Analysis

Given a set of zero-mean observations x_k, k = 1...M, $x_k \in R^N$, and $\sum_{k=1}^{M} x_k = 0$, the covariance matrix is

$$C = \frac{1}{M}\sum_{j=1}^{M} x_j x_j^T \tag{3.2.1}$$

PCA aims to find the projection direction that maximizes the variance, which is equivalent to finding the eigenvalue from the covariance matrix

$$\lambda w = Cw \tag{3.2.2}$$

for eigenvalues $\lambda \geq 0$ and $w \in R^N$. Since $Cw = \frac{1}{M}\sum_{j=1}^{M}(x_j \cdot w)x_j$, all solutions **w** with $\lambda \neq 0$ must lie in the span of $x_{1, \ldots,} x_M$. Therefore

$$\lambda(x_k \cdot w) = (x_k \cdot C\ w), k = 1,..., M \tag{3.2.3}$$

In Kernel PCA, each vector x is projected from the input space, R^N, to a higher dimensional feature space, R^F, by a nonlinear map:

$$\Phi : R^N \rightarrow R^F, F >> N \tag{3.2.4}$$

Dimensionality of the feature space can be arbitrarily large. In R^F, the covariance matrix of $\Phi(x)$ is

$$C^{\Phi} = \frac{1}{M}\sum_{j=1}^{M} \Phi(x_j)\Phi(x_j)^T \tag{3.2.5}$$

and the corresponding eigenvalue problem is

$$\lambda w^{\Phi} = Cw^{\Phi} \tag{3.2.6}$$

All solutions w^{Φ} with $\lambda \neq 0$ lie in the span of $\Phi(x_1),...,\Phi(x_M)$.

$$\lambda(\Phi(x_k)\cdot w^{\Phi}) = (\Phi(x_k)\cdot Cw^{\Phi}) \qquad k = 1,...,M \tag{3.2.7}$$

and w^{Φ} lie in the span of $\Phi(x_1),...,\Phi(x_M)$. such that

$$w^{\Phi} = \sum_{i=1}^{M} \alpha_i \Phi(x) \tag{3.2.8}$$

Using Equations (3.2.7) and (3.2.8), we have, for k = 1,…,M,

$$\lambda \sum_{i-1}^{M} \alpha_i (\Phi(x_k)\cdot\Phi(x_i)) = \frac{1}{M}\sum_{i=1}^{M}\alpha_i(\Phi(x_k)\cdot\sum_{i-1}^{M}\Phi(x_j)(\Phi(x_j)\cdot\Phi(x_i)) \tag{3.2.9}$$

Defining an *M x M* matrix *K* by

$$K_{ij} = k(x_i, x_j) = \Phi(x_j)\cdot\Phi(x_i) \tag{3.2.10}$$

Equation (3.2.9) can be rewritten as

$$M\lambda K\alpha = K^2\alpha \tag{3.2.11}$$

where α denotes a column vector with entries $\alpha_1,...,\alpha_M$. The solutions of Equation (3.2.11) is the same to the following eigenvalue problem,

$$M\lambda\alpha = K\alpha \tag{3.2.12}$$

Boser, Guyon and Vapnik suggested the use of Gaussian Radial Basis Function kernel [46]

$$k(x_i, x_j) = \exp(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}) \tag{3.2.13}$$

In this thesis, since the size of the training images do not let the computation of Radial Basis Function kernel, mean square method is used for transformation into higher space, i.e.

$$k(x_i, x_j) = \left\| x_i - x_j \right\|^2 \tag{3.2.14}$$

This decreased the performance of the method; however recognition rate is still better than the Eigenface method.

Conventional PCA is a special case of Kernel PCA with polynomial kernel of first order. In other words, Kernel PCA is a generalization of conventional PCA since different kernels can be utilized for different nonlinear projections.

The vectors can now be projected in R^F to a lower dimensional space spanned by the eigenvectors w^Φ, x is a test sample whose projection is $\Phi(x)$ onto the eigenvectors w^Φ are the nonlinear principal components corresponding to Φ

$$w^\Phi \cdot \Phi(x) = \sum_{i=1}^{M} \alpha_i (\Phi(x_i) \cdot \Phi(x)) = \sum_{i=1}^{M} \alpha_i k(x_i, x) \tag{3.2.14}$$

In other words, the first q $(1 \le q \le M)$ nonlinear principal components can be extracted using the kernel function without the expensive operation to explicitly project the patterns to a high dimensional space R^F. The first q components correspond to the first q non-increasing eigenvalues of Equation (3.2.12).

Properties of Kernel PCA

Several properties of Kernel PCA is discussed in terms of feature extraction and reconstruction in this section.

Dimensionality and Feature Extraction

Kernel PCA method can extract more principal components than linear PCA. Consider a problem consisting of *M* observations x where the dimension of x is *N* and *M >> N*. Linear PCA can find at most *N* nonzero eigenvalues from the covariance matrix($C = \frac{1}{M} \sum i = 1^{M} x_i x^T$). In contrast, Kernel PCA can find up to *M* nonzero eigenvalues from the covariance matrix $C^{\Phi} = \frac{1}{M} \sum_{i=1}^{M} \Phi(x_i) \Phi(x_i^T)$ where Φ is a nonlinear mapping function that can project x_i to a possibly infinite-dimensional feature space.

Reconstruction

Since PCA is essentially a basis transformation, each pattern can be exactly reconstructed using all the principal components and the basis vectors (i.e., eigenvectors).

In contrast, there is no direct counterpart in Kernel PCA. Due to nonlinear mapping, a vector in high dimensional feature space does not necessarily have a pre-image in the input space. An approximate reconstruction of the image of a pattern in R^F can be best found from its nonlinear components. This can be achieved by a regression method for estimating the mapping from kernel-based principal components to the input space.

3.3 Face Recognition by Subspace LDA Method

Subspace LDA method is simply the implementation of PCA by projecting the data onto the eigenface space and then implementing LDA to classify the eigenface space projected data. Projecting the data to the eigenface space *generalize*s the data, whereas implementing LDA by projecting the data to the classification space *discriminate*s the data. Thus, Subspace LDA approach seems to be a complementary approach to the Eigenface method.

PCA and LDA were described in the previous sections, hence here just the implementation is described.

The image I is represented by (N_x x N_y) pixels.

$$(N_x \text{ x } N_y) \text{ pixels} \tag{3.3.1}$$

The image matrix I of size (N_x x N_y) pixels is converted to the image vector of size (P x 1) where P = (N_x x N_y) . Training set Γ is set of image vectors and its size is (P x M_t) where M_t is the number of the training images:

$$\Gamma = \left[\Gamma_1 \Gamma_2 ... \Gamma_{Mt} \right] \tag{3.3.1}$$

Mean face Ψ is the arithmetic average of the training image vectors at each pixel point and its size is (P x 1):

$$\Psi = \frac{1}{M_t} \sum_{i=1}^{M_t} \Gamma_i \tag{3.3.3}$$

Mean subtracted image Φ is the difference of the training image from the mean image (size P x 1):

$$\Phi = \Gamma - \Psi \tag{3.3.4}$$

Difference Matrix A is the matrix of all the mean subtracted training image vectors and its size is (P x M_t):

$$A = [\Phi_1 \Phi_2 \Phi_3 ... \Phi_{Mt}\ ] \tag{3.3.5}$$

Covariance Matrix X is the covariance matrix of the training image vectors of size (P x P):

$$X = A \cdot A^T = \frac{1}{M_t} \sum_{i=1}^{M_t} \Phi_i \Phi_i^T \tag{3.3.6}$$

The training image vectors can be projected to the eigenface space and thus the weight of each eigenvector to represent the image in the eigenface space is calculated;

Projection w_k is the representation of the training image in the eigenface space and its size is (M' x 1):

$$w_k = v_k^T \cdot \Phi = v_k^T \cdot (\Gamma - \Psi) \tag{3.3.7}$$

Weight Matrix (M' x 1) Ω is:

$$\Omega = \left[w_1 \ \ w_2 \ ... \ w_{M'} \right]^T \tag{3.3.8}$$

By performing all these calculations, the training images are projected onto the eigenface space; that is a transformation from P dimensional space to M' dimensional space. This PCA step is achieved to reduce the dimension of the data, also may be referred as a feature extraction step. From this step on, the each image is an (M' x 1) dimensional vector in the eigenface space. With the projected data, a new transformation is performed; the classification space

projection by LDA. Instead of using the pixel values of the images (as done in pure LDA), the eigenface projections are used in the Subspace LDA method. Again, as in the case of pure LDA, a discriminatory power $J(T)$ is defined as;

$$J(T) = \frac{\left| T^T \cdot S_b \cdot T \right|}{\left| T^T \cdot S_w \cdot T \right|} \tag{3.3.9}$$

where S_b is the between-class and S_w is the within-class scatter matrix. For c individuals having q_i training images in the database, the within-class scatter matrix S_w is computed as;

$$S_w = \sum_{i-1}^{c} P(C_i) \sum_i \tag{3.10}$$

which represents the average scatter $\sum_i$ of the projection Ω in the eigenface space of different individuals C_i around their respective means m_i. The size of S_w depends on the size of the eigenface space; if M' of the eigenfaces were used, then the size of S_w is (M' x M').

Here, eigenface space class mean m_i is defined as;

$$m_i = \frac{1}{q_i} \sum_{k=1}^{q_i} \Omega_k \tag{3.3.11}$$

is the arithmetic average of the eigenface projected training image vectors corresponding to the same individual; i = 1, 2, …, c and its size is (M' x 1).

Moreover, mean face m_0 is calculated from the arithmetic average of all the projected training image vectors:

$$m_0 = \frac{1}{M_t}\sum_{k=1}^{M_t}\Omega_k \tag{3.3.12}$$

The average scatter $\sum_i$ is calculated as:

$$\sum_i = E\left[(\Omega - m_i)\cdot(\Omega - m_i)^T\right] \tag{3.3.13}$$

Also between-class scatter matrix S_b is computed as;

$$S_b = \sum_{i=1}^{c} P(C_i)(m_i - m_0)\cdot(m_i - m_0)^T \tag{3.3.14}$$

which represents the scatter of each projection classes mean m_i around the overall mean vector m_0 and its size is (M' x M'). $P(C_i)$ is the prior class probability and may be written as

$$P(C_i) = \frac{1}{c} \tag{3.3.15}$$

with the assumption that each class has equal prior probabilities. The objective is to maximize *J(T)*; that is, to find an optimal projection *W* which maximizes between-class scatter and minimizes within-class scatter.

$$W = \arg\max_T(J(T)) \Rightarrow \max(J(T)) = \frac{\left| T^T \cdot S_b \cdot T \right|}{\left| T^T \cdot S_w \cdot T \right|}\Big|_{T=W} \tag{3.3.16}$$

Then, *W* can be obtained by solving the generalized eigenvalue problem;

$$S_b W = S_w W \lambda_w \tag{3.3.17}$$

Next, the eigenface projections of the training image vectors are projected to the classification space by the dot product of optimum projection *W* and weight

vector. Classification space projection $g(\Omega_i)$ which is the projection of the training image vectors' eigenface projections to the classification space of size (*(c-1)* x 1) where i = 1, 2… M_t is:

$$g(\Omega_i) = W^T \cdot \Omega_i \tag{3.3.18}$$

The training stage is completed in this step.

Test image vector:

$$\Gamma_T \tag{3.3.19}$$

is the test image vector of size (P x 1).

Mean subtracted image Φ_T is the difference of the test image from the mean image (size P x 1):

$$\Phi_T = (\Gamma_T - \Psi) \tag{3.3.20}$$

Projection w_k is the projection of a training image on each of the eigenvectors where k = 1, 2… M':

$$w_k = v_k^T \cdot \Phi_T = v_k^T \cdot (\Gamma_T - \Psi) \tag{3.3.21}$$

Weight Matrix Ω_T is the representation of the test image in the eigenface space and its size is (M' x 1):

$$\Omega_T = \left[w_1 \;\; w_2 \;\; ... \; w_{M'} \right]^T \tag{3.3.22}$$

Then, the eigenface projection of the test image vector (i.e. weight matrix) is projected to the classification space in the same manner. Classification space projection $g(\Omega_T)$ which of size (*(c-1)* x 1) is:

$$g(\Omega_T) = W^T \cdot \Omega_T \tag{3.3.23}$$

Finally, the distance between the projections is determined by the Euclidean distance d_{Ti} between the training and test classification space projections measure which is scalar and calculated for i = 1, 2, …, M:

$$d_{Ti} = \left\| g(\Omega_T) - g(\Omega_i) \right\| = \sqrt{(g_k(\Omega_T) - g_k(\Omega_i))^2} \tag{3.3.24}$$

Some other distance measures (strictly weighted and softly weighted) are also suggested, but that measures were not studied in this thesis.

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter, the implemented algorithms are tested individually. After testing, these methods' performances are compared on three different face databases:

- Yale
- AT&T
- Metu Vision face databases.

The tests are conducted via writing programs in Matlab and Java Programming Language. Training part is implemented in Matlab, since most Mathematical functions are available and testing procedures are implemented in Java because of the implementation time consideration and usability through internet.

4.1 Face Databases Used for Testing

4.1.1 Yale Database

The Yale database contains 165 frontal face images of 15 individuals taken with variation both in facial expression and lighting. Yale face database can be seen in Appendix A.

4.1.1 AT&T Database (Formerly ORL Database)

The AT&T (Formerly Olivetti and Oracle Research Laboratory) face database is used in order to test Eigenface, Kernel Eigenface and Subspace LDA methods in the presence of head pose variations. There are ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying lighting, facial expressions (open / closed eyes, smiling / not smiling), facial details (glasses / no glasses) and head pose (tilting and rotation up to 20 degrees). All the images were taken against a dark homogeneous background. In Appendix B the whole set of 40 individuals 10 images per person from the AT&T database are displayed.

In the following sections various tests are conducted with this database, such as changing the training and test images used for each individual and using less or more number of training and test images in order to test the systems reaction to these changes.

4.1.1 Metu Vision Face Database

The Metu Vision Lab Face Database is collected during this thesis study, in order to test the systems performances in different poses and expressions of faces. The images are taken from 3 meters distance with various poses and expressions of the individuals. In Appendix C whole set of 15 individuals 10 images per person from the Metu Vision Lab Face Database can be seen. Naming convention is in the form xxx1 where **x** ranges from 001 to 150.

The files are in jpeg (Joint Photographics Experts Group) file format. The size of each image is 76 x 106 (width x height), 8-bit grey levels.

Since there are 10 images per individual in the Metu Vision Lab database, a number of them are used to train the systems and the rest are used to test them. Performances of the methods in this thesis are tested against different angles of poses. Additionally, changing the training and test images used for each individual and using less number of training and test images in order to test the systems reaction to these changes are also tested in the following sections.

4.2 Experimental Results

4.1.1 Experiments Performed with the Yale Database

4.2.1.1 Eigenface Method:

In Figure 4.1 there is a chart of the variance captured by each principal component, where we assume that the variance for k components is equal to the cumulative sum of those components, divided by the total sum. For k=10, variance= 0.7586. For k=25, variance = 0.9352.

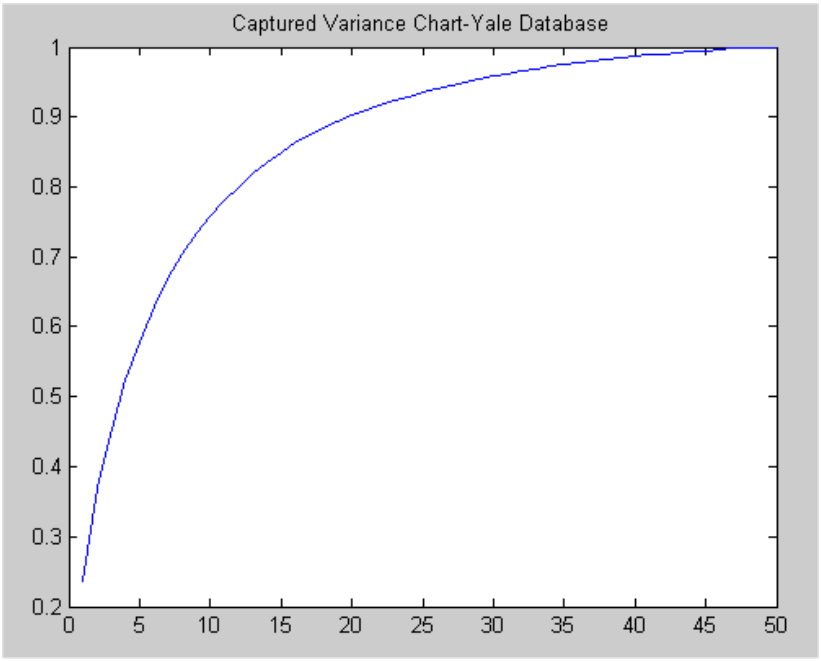


Figure 4.1: Captured Variance Chart

In Figure 4.2 the Eigenvalue spectrum of Yale Database can be seen. It should be noticed that Eigenvalue spectrum of any database is independent of number of eigenvectors but when new images are inserted into the database the spectrum changes due to the change in covariance matrix. Figure 4.3 shows the mean face obtained from the AT&T database.

In Figure 4.4, each new image from left to right corresponds to using 1 additional principle component for reconstruction. As it can be seen, the figure becomes recognizable around the 7th or 8th image, but not perfect.

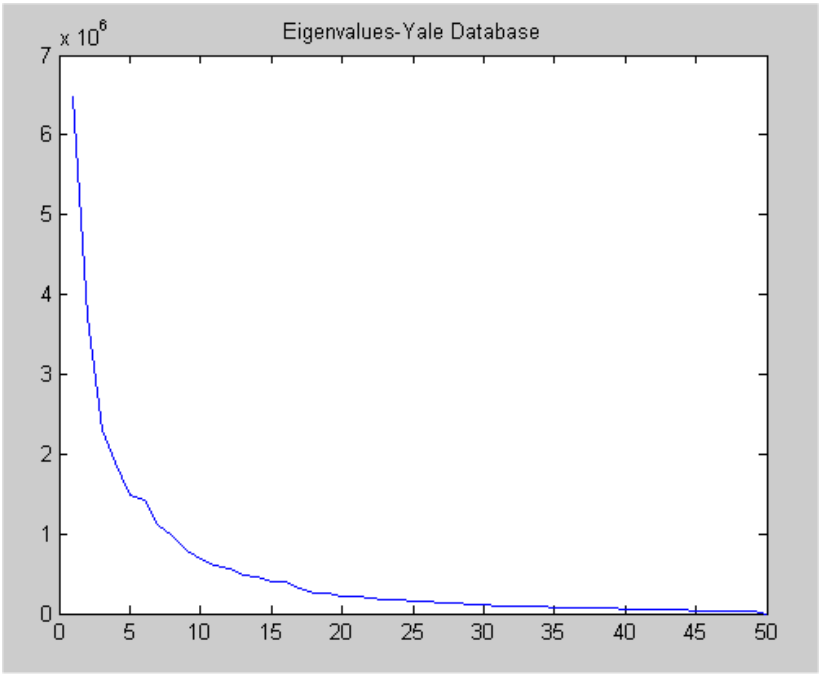


Figure 4.2: Eigenvalue spectrum of Yale database

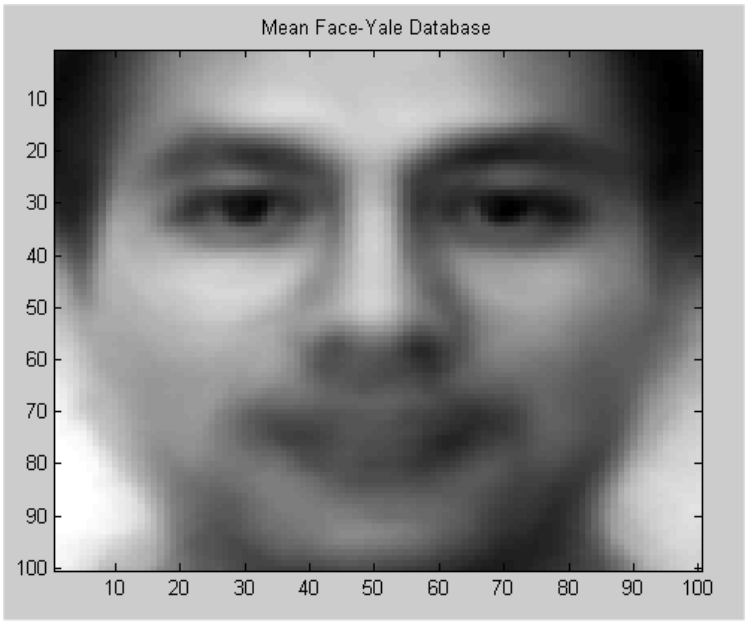


Figure 4.3: Mean Face-Yale Database

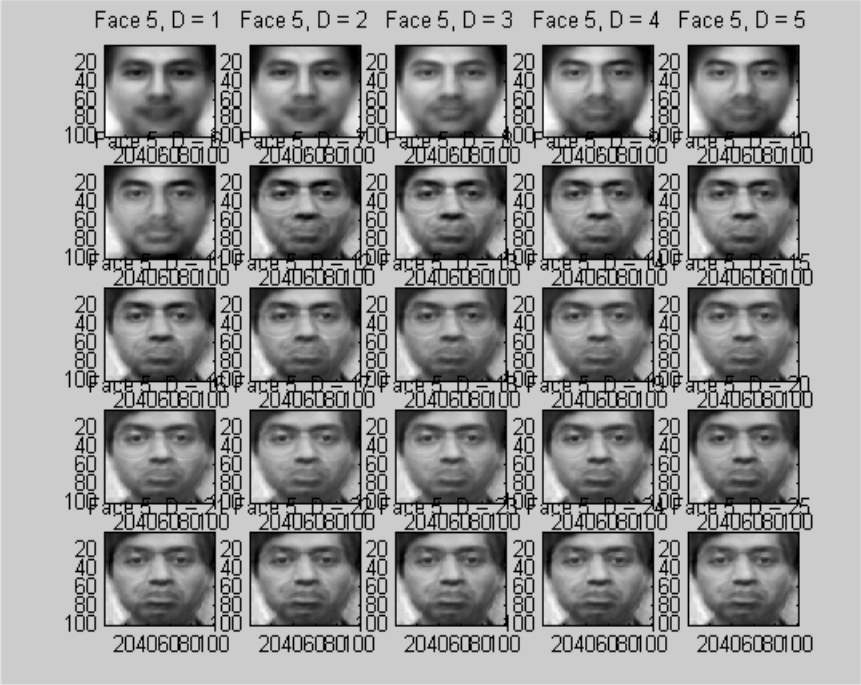


Figure 4.4: Reconstruction of an image with both lighting variations and glasses

In Figure 4.5, the reconstruction results when excluding the images with glasses can be seen. It is a similar reconstruction to the one in Figure 4.4. The images converge to the correct face slightly faster, but not by much.

In Figure 4.6 the images where the dataset excludes all those images with either glasses or different lighting conditions are displayed. The point to keep in mind is that each new image represents one new principle component. As it can be seen, the image converges extremely quickly.

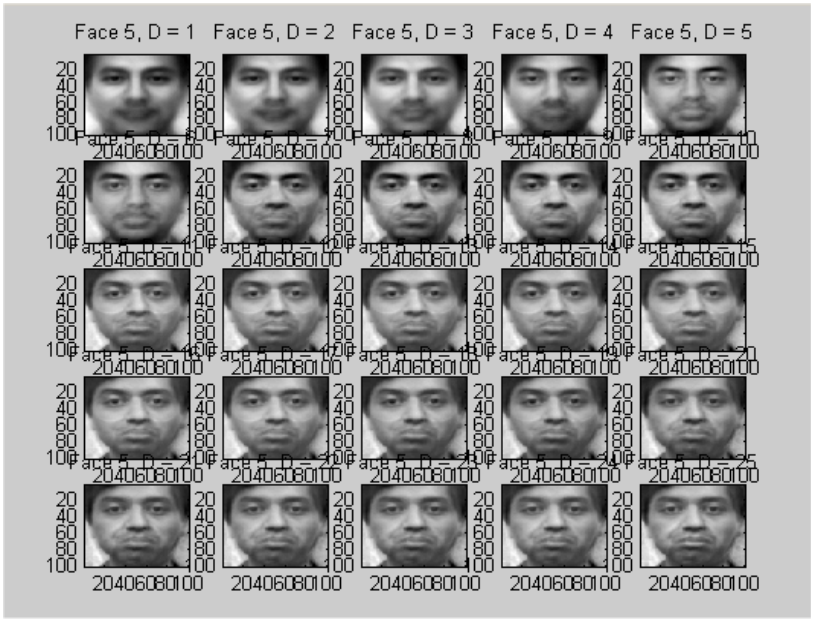


Figure 4.5: Reconstruction of an image with lighting variations, no glasses

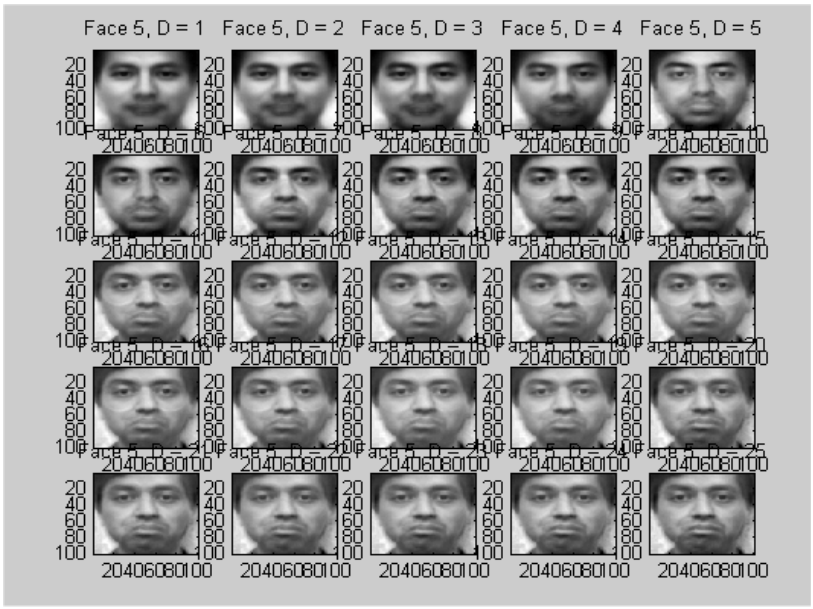


Figure 4.6: Reconstruction of an image with no lighting variations, no glasses

In order to visualize the difference in recognition rate corresponding to changes in lighting and changes in expression, some recognition experiments on subsets of the database are tried by changing the images of each individual used in the training and testing stage. These test results can be observed in Table 4.1.

In the first raw of Table 4.1, recognition without the glasses-wearing individuals can be seen. The results are slightly better, but overall, it seems that glasses do not make a major impact in recognition.

Secondly, Table 4.1 shows recognition rate where only the normal lighting and no glasses images are used (except for the people that are always wearing glasses). The recognition is much better.

It is expected that, the success rates would be closer to each other and the results would be more stable if there were much more images per individual.

Table 4.1: Performances of Eigenface method using various training and test images per individual (Yale Database)

Property of Training Images	Training Images	Test Images	Performance
No glasses images	1-3-4-5-6	2-7-8-9-10-11	80.0%
Normal lighting & no glasses images	1-3-5-6-8	2-4-7-9-10-11	82.0%
Variation in lighting & expression	1-2-3-4-5-	6-7-8-9-10-11	81.0%

According to the recognition experiments on subsets of the database corresponding to changes in lighting and changes in expression, it is much easier

to accommodate a change in expression. Presumably this is because the relative intensities of the pixels are very similar between images of different expressions, while those with different lightings display very different intensities. This means the eigenfaces technique is robust against local changes, but not to global changes.

In Figure 4.7 and Figure 4.8 recognition samples can be seen. The lighting has been varied and in Figure 4.7, however in Figure 4.8**,** only the expressions are different.

About %30 of all of the classifications are false recognitions in Figure 4.8. All of the classifications are correct in Figure 4.8 where the expressions are different.

Figure 4.7: Recognition Examples (Lighting Variation)

Figure 4.8: Recognition Examples (Expression Variation)

4.2.1.2 Kernel Eigenface Method:

Simulation results for Kernel Eigenface Method using Yale database as a training set can be seen in Table 4.2.

Table 4.2: Performances of Kernel Eigenface method using various training and test images per individual (Yale Database)

Property of Training Images	Training Images	Test Images	Performance
No glasses images in training set	1-3-4-5-6	2-7-8-9-10-11	%81.82
Normal lighting & no glasses images in training set	1-3-5-6-8	2-4-7-9-10-11	%85
Variation in lighting & expression in training set	1-2-3-4-5-	6-7-8-9-10-11	%84.55

4.2.1.3 Subspace LDA Method:

Simulation results for Subspace LDA method using Yale database as a training set can be seen in Table 4.3.

Table 4.3: Performances of Fisher LDA method using various training and test images per individual (Yale Database)

Property of Training Images	Training Images	Test Images	Performance
No glasses images	1--3-4-5-6	2-7-8-9-10-11	85.5%
Normal lighting & no glasses images	1-3-5-6-8	2-4-7-9-10-11	89%
Variation in lighting & expression	1-2-3-4-5-	6-7-8-9-10-11	90.3%

4.2.2 Experiments Performed with the AT&T Database

4.2.2.1 Eigenface Method:

The AT&T (Formerly Olivetti and Oracle Research Laboratory) face database is used in order to test Eigenface method in the presence of head pose variations. There are ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying lighting, facial

expressions (open / closed eyes, smiling / not smiling), facial details (glasses/no glasses) and head pose (tilting and rotation up to 20 degrees). All the images were taken against a dark homogeneous background. When the first images for each 40 individuals are taken as reference and the rest is used for testing purposes, Eigenface method achieved 70% correct classification with AT&T database. It is observed that the performance of the method decreased slightly due to the orientation in depth of the head.

In Figure 4.9 erroneously recognized faces of AT&T database are presented. These results were expected, since locations of the facial features (eyes, nose, mouth….) are quite changing with rotation of the head.

(a)

(b)

Figure 4.9: (a) Erroneously classified faces of AT&T database when number of eigenvectors is 20
(b) Erroneously classified faces of AT&T database when number of eigenvectors is 10

In Figure 4.10, first 20 eigenfaces are shown. It can be seen from the figure that the eigenvectors that have the most significant eigenvalues are much similar to a face.

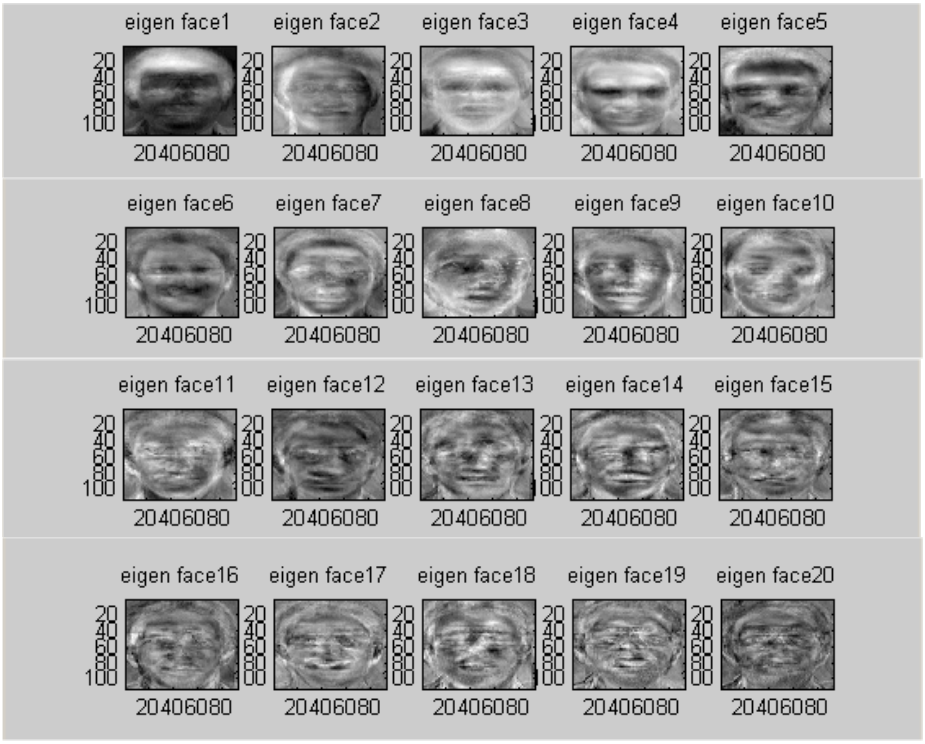


Figure 4.10: Eigenfaces sorted with respect to their eigenvalues

In Figure 4.11, it can be observed that most of the generalization power is contained in the first few eigenvectors. For example, 40% of the total eigenvectors have 85 – 90% of the total generalization power. Thus, using 40% of the total number of eigenvectors may end up with reasonable classification results.

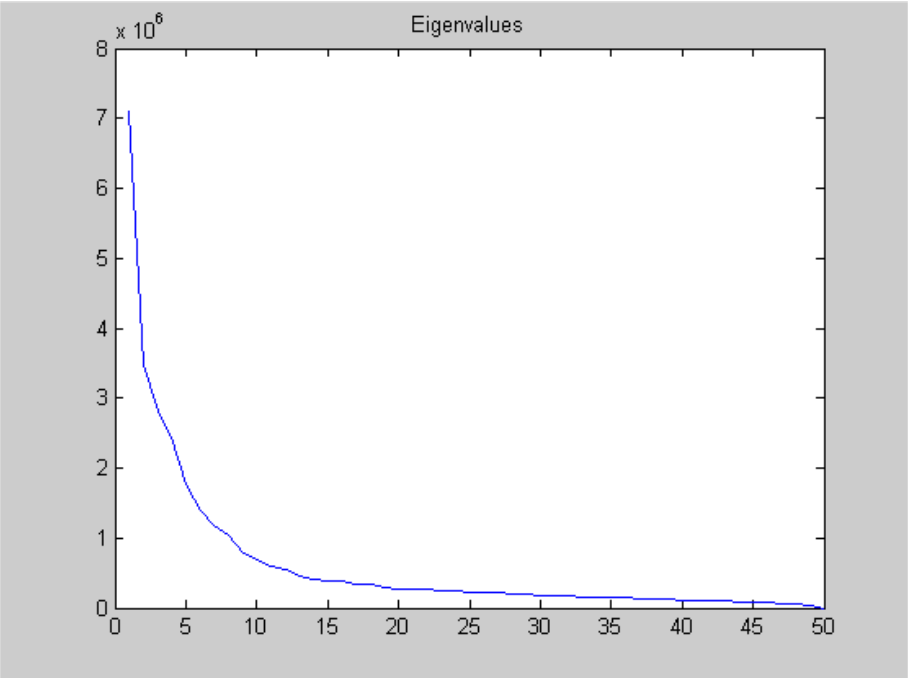


Figure 4.11: Eigenvalue spectrum of AT&T database

In Figure 4.12 mean face obtained from the AT&T database can be seen.

Recognition performances using different number of training and test images can be observed in Table 4.4. Eigenface method achieves best correct recognition rate by using equal number of reference facial images and test images for each individual.

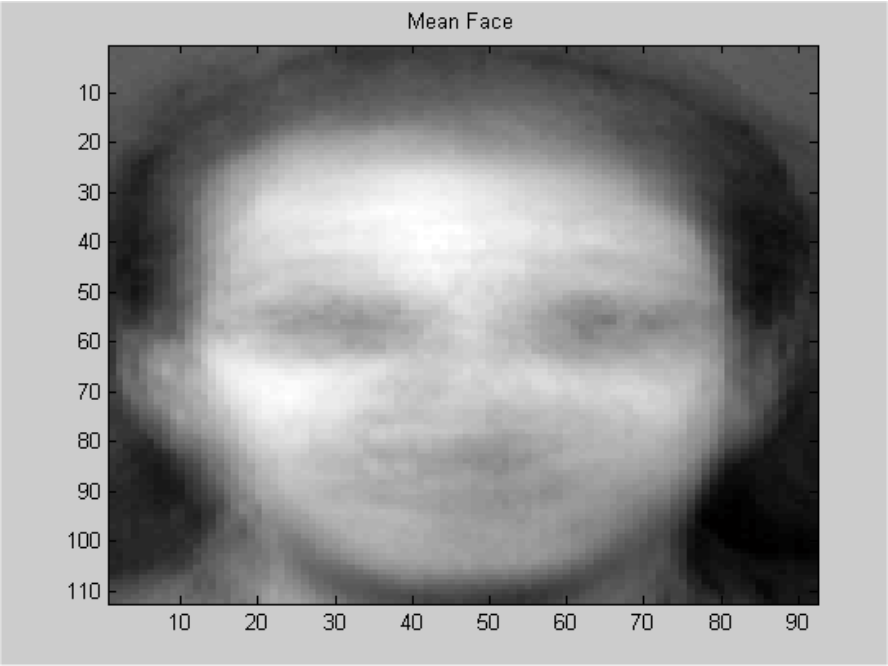


Figure 4.12: Mean Face of AT&T database (112x92)

Table 4.4: Performance of Eigenface method using various numbers of training and test images. (AT&T Database)

Total Number of Images	Number of Training Images (Per Individual)	Number of Test Images (Per Individual)	Success Rate
400	1	9	%70
400	5	5	%82
400	9	1	%80

The next experiment is done with different number of eigenvectors, from 5 to 50. The eigenvectors are chosen according to their significance, i.e. according to the corresponding eigenvalues (the larger the eigenvalue, the more

significant). For example, if the number of eigenvectors to be tested is 15, then, the 1st most significant to the 15th most significant eigenvectors are chosen.

Figure 4.13 shows that the more eigenvectors are chosen, the higher the recognition rate becomes. With the same background and nearly the same illumination, the eigenvectors together characterize the variations mainly due to the features of the faces of different identity in the training set. With more eigenvectors, more features of faces can be achieved to reconstruct an image that is more similar to the original image. Therefore, the recognition rate is higher. From the test result, it shows that the performance is quite stable, around 80%, when the number of eigenvectors is around 25 to 30. The reason is that not all eigenvectors contribute equal amount to represent a testing image. Some eigenvectors contribute more and some contribute less. Therefore, in some situation, if the number of eigenvectors is increased by one or two, the additional eigenvectors may not have much contribution. In other words, sufficient information is got. With just a little bit more information, it may not be useful. It is because the little information is not enough to help more testing images to be identified successfully. For Eigenface, there is a trade off between the number of eigenvectors and speed. With more eigenvectors, the error rate is lower, but the speed is lower too. With less eigenvectors, the speed is higher, but the error rate is higher too. In the test, when the number of eigenvectors is from 25 to 30, the recognition performance is similar, around 80%. Therefore, if one accepts a

recognition performance of 80%, then, 25 eigenvectors should be chosen in order to have a higher speed.

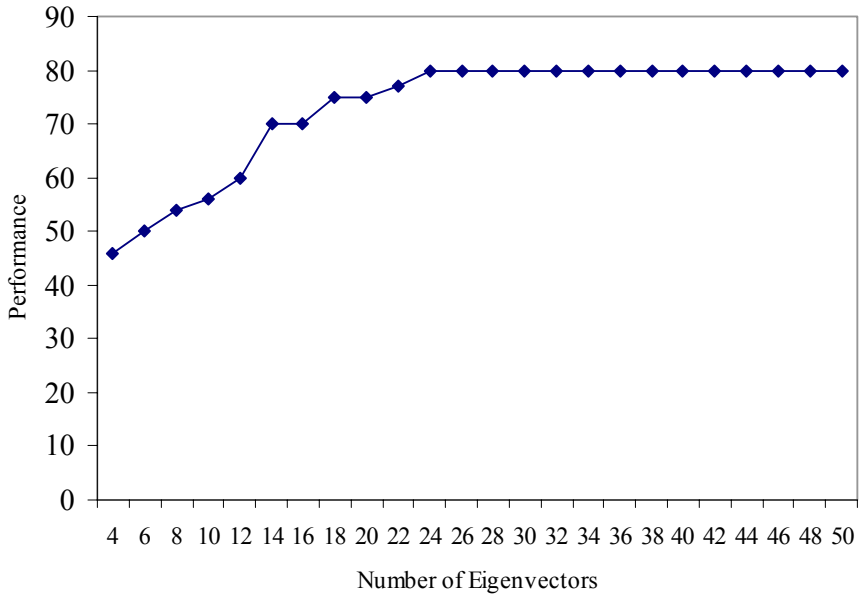


Figure 4.13: Performance With Respect to Number of Eigenvectors (Eigenface Method-AT&T Database)

Applying histogram equalization as a preprocessing step did not cause much change in the AT&T database.

4.2.2.2 Kernel Eigenface Method:

Using 5 training images and 5 testing images with the same number of eigenvectors (30), Kernel PCA methods achieve lower error rates than the Eigenface approach on the AT&T dataset. Table 4.6 shows the recognition rates of both methods in the conditions specified above:

Table 4.6: Experimental results on AT&T database withKernel PCA

Method	Reduced space	Performance (%)
Eigenface	30	82
Kernel PCA	30	85

In Figure 4.14 erroneously recognized faces of AT&T database are presented. These results were expected, since locations of the facial features (eyes, nose, mouth….) are quite changing with rotation of the head. Also a similarity between individuals and lighting conditions play a role in these results.

Figure 4.14: Erroneously classified faces of AT&T database with Kernel Eigenface Method

4.2.2.3 Subspace LDA Method:

Similar test procedures are used in Subpace LDA method with the Eigenface method to have realistic comments for the comparison of both

methods. At first, different numbers of training and testing images are used to visualize performance in the various numbers of training images.

As it can be observed from Table 4.7 that, recognition performance is slightly better than Eigenface method in Table 4.1 But in real-time applications Eigenface method is more advantageous than the Subpace LDA method, since training step in Subpace LDA also includes the scatter matrix calculations.

Table 4.7: Performance due to variation in number of training& testing images

Total Number of Images	Number of Training Images (Per Individual)	Number of Test Images (Per Individual)	Success Rate
400	1	9	74.5%
400	5	5	85%
400	9	1	83%

For Eigenface, one photo for each person is enough for training. However, for Subspace LDA, it requires several images for training, which are not always available for some applications. So, Eigenface method is again more useful for real life like suspect recognition. One solution to this problem for Subspace LDA method may be to derive multiple samples from a single face image, by which Subspace LDA can be trained.

In Figure 4.15, performance of Subspace LDA with respect to number of Eigenvectors can be observed. Since, Subspace LDA is implemented we have

also PCA technique in this method. As it can be observed from Figure 4.15 that performance is higher when Eigenvector number is over 40 and it is very low when Eigenvector number is between 20 and 30. When compared to Eigenface method, Subspace LDA has higher performance when we have sufficient Eigenvectors to separate the classes used in training.

Using Histogram Equalization preprocessing technique slightly decreased the performance.

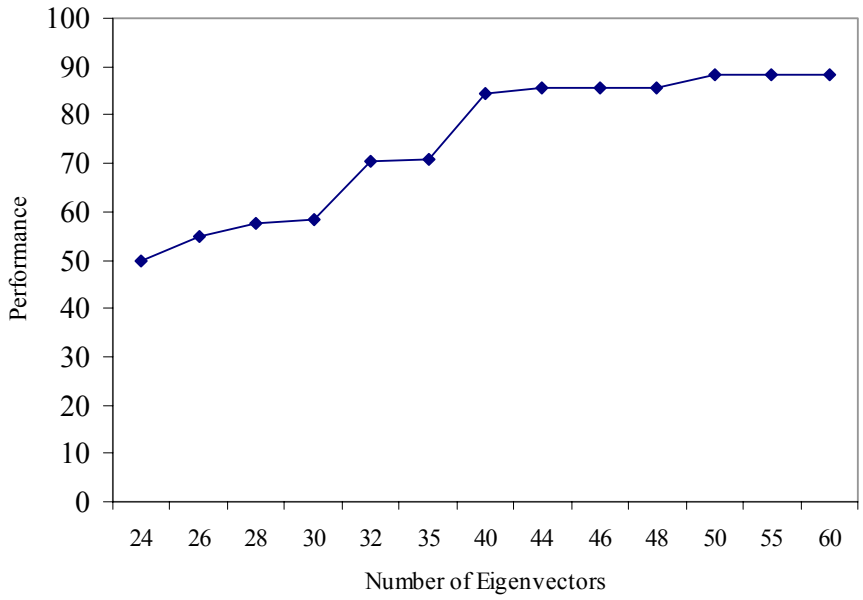


Figure 4.15: Performance in AT&T Face Database With Respect to Number of Eigenvectors in (Subspace LDA)

4.2.2 Experiments Performed with the METU Vision Face Database

4.2.3.1 Eigenface Method:

The effect of using different number of eigenvectors while projecting the images (i.e. using Eigenfaces with different dimensions) is studied. When the Eigenface space's dimension is small (4-8) performance is low and when the Eigenface space's dimension is large (15-20) performance becomes better. Since the dimension of METU Vision database is smaller than AT&T database 15 eigenvectors are enough to separate the face classes used in training. Necessary eigenvector number was 25 in AT&T face database to catch the same performance with the METU Vision database. In Figure 4.16 erroneously recognized faces are shown. From Figure 4.18 the figure it can be seen that a small amount of rotation in head (15º-20º) or a small change in facial features (taking off glasses) may be a problem for face recognition. The images in METU Vision database are converted into grayscale before training and testing.

Figure 4.16 Erronously recognized faces in METU Vision Database with Eigenface method.

Figure 4.16 Erronously recognized faces in METU Vision Database with Eigenface method. (cont'd)

Performance of Eigenface method can be observed in Figure 4.17 for various numbers of eigenvectors in METU Vision database.

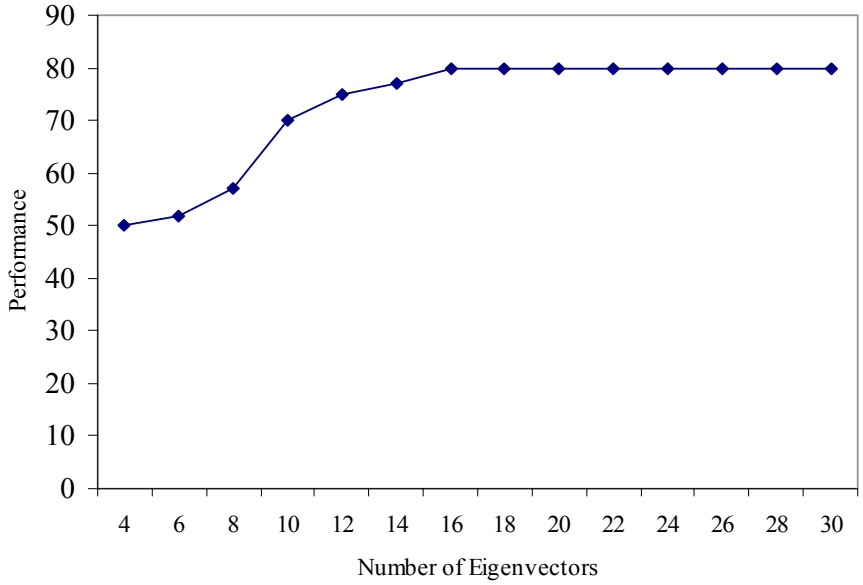


Figure 4.17: Performance With Respect to Number of Eigenvectors (Eigenface Method-METU Vision Database)

The training set consists of 10 face images of 15 different persons in Metu Vision. 5 images are used for training and 5 are used for testing so, there are 75 testing images of different facial expression. No testing image is exactly the same as one of the images in the training set. All face images in the training set and the testing images are of the same size and have the same background. Performance of the Eigenface algorithm against change in illumination is measured with using different number of eigenvectors. Eigenvector numbers are selected to be 15, 20, 25, 30 and 40. The eigenvectors are chosen according to their significance in each case, i.e. from the 1st most significant to the 15th most significant eigenvectors when the number of eigenvectos are chosen to be 15 and from the 1st most

significant to the 20th most significant eigenvectors when the number of eigenvectos are chosen to be 20 etc. The experiment is done by changing the brightness of all 75 testing images by –20%, -10%, 0%, +10% and +20%. Test results for this experiment can be seen in Tables 4.8 (a), (b), (c), (d), and (e).

Table 4.8: Performance due to different brightness on the testing images

(a) (Eigenface) - # of Eigenvectors: 15

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	18.6667	61.33%	78.66%	61.33%	17.33%
Histogram Equalized	65.33%	69.33%	77.33%	69.33%	69.33%

(b) (Eigenface) - # of Eigenvectors: 20

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	18.66%	65.33%	80.0%	62.66%	18.66%
Histogram Equalized	65.33%	69.33%	78.66%	69.33%	69.33%

(c) (Eigenface) - # of Eigenvectors: 25

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	20%	66.66%	80.0%	65.33%	18.66%
Histogram Equalized	69.33%	69.33%	81.33%	69.33%	69.33%

(d) (Eigenface) - # of Eigenvectors: 30

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	20%	66.66%	80.0%	66.66%	18.66%
Histogram Equalized	69.33%	70.66%	78.66	70.66%	70.66%

(e) (Eigenface) - # of Eigenvectors: 40

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	20%	66.66%	80.0%	65.33%	18.66%
Histogram Equalized	69.33%	69.33%	78.66	69.33%	70.66%

With no additional change in the brightness, the error rate is just 20%. When the brightness is increased or decreased by 10%, the error rate is nearly doubled. With 20% increasing or decreasing in brightness, the error rate is tripled, which is unacceptable. When histogram equalization is used algorithm becomes more robust against illumination change. This test shows that brightness variation on the testing images severely affects the error rate. Therefore, for Eigenface, in order to increase the performance, the images to be tested should have equal or nearly the same brightness with the images in the training set or histogram equalization should be used as a preprocessing step.

4.2.3.2 Kernel Eigenface Method:

Same strategy is used with Eigenface method while testing Kernel Eigenface method against different number of eigenvector with Metu Vision face database.

Performance is slightly better than Eigenface method. As shown in Figure 4.18, when the Eigenface space's dimension is small (4-8), performance is low and when the Eigenface space's dimension is large (15-20). But, in each case recognition rate is slightly better than Eigenface method.

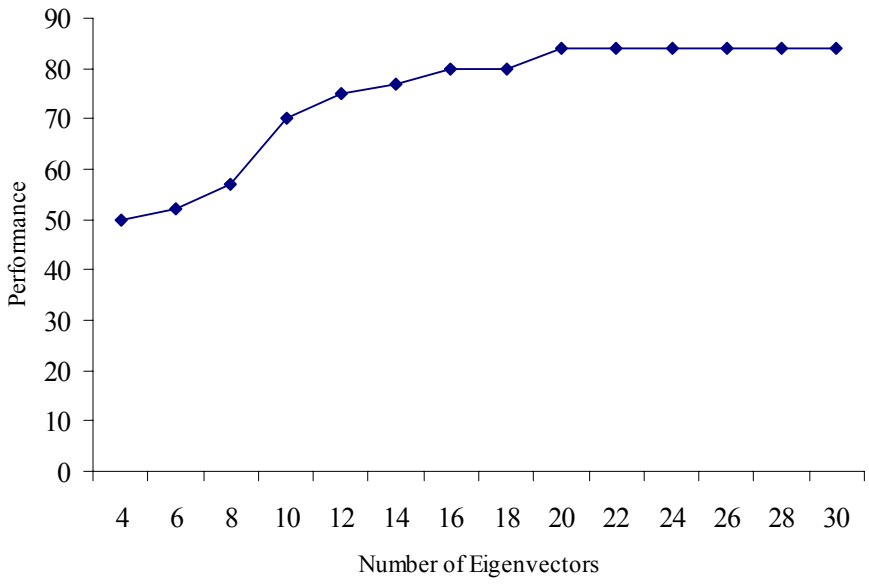


Figure 4.18: Performance With Respect to Number of Eigenvectors in (Kernel Eigenface-METU Vision database)

Performance decreases by changing brightness of the training images but it is not as low as in eigenface method. This is an expected result because Kernel PCA is a subspace method that uses nonlinear function for transformation. By this way, recognition task is not affected by facial expression variations, illumination changes image background differences and other complications

Performance of the Kernel Eigenfaces due to brightness variation is shown in Tables 4.9 (a), (b), (c), and (d). It can be seen from the results in Table 4.9 that Kernel Eigenface method is more robust against illumination changes

with respect to Eigenface method. Applying histogram equalization as a preprocessing step in this method does not affect the performance so much.

Table 4.9: Performance due to different brightness on the testing images

(a) (Kernel Eigenface) - # of Eigenvectors: 10

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	18.66%	44%	69.33	26.66%	16%
Histogram Equalized	%58.66	64%	68.0%	66.66%	65.33%

(b) (Kernel Eigenface) - # of Eigenvectors: 15

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	69.33%	69.33%	78.66%	30.66%	16%
Histogram Equalized	64%	66.66%	81.33%	68%	69.33%

(c) (Kernel Eigenface) - # of Eigenvectors: 20

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	65.3%	66.6%	84.0%	70.6%	66.6%
Histogram Equalized	66.66%	69.33%	81.33%	70.6%	70.66%

(d) (Kernel Eigenface) - # of Eigenvectors: 25

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	66.6%	68.0%	84.0%	66.6%	66.6%
Histogram Equalized	69.33%	72.0%	81.33%	73.33%	73.33%

4.2.3.3 Subspace LDA Method:

Performance of Subspace LDA method is slightly better than Eigenface method. As shown in Figure 4.19, when the Eigenface space's dimension is small (7-10) performance is low because sufficient number of eigenvectors is necessary to separate the classes in training for Subpace LDA method. When the Eigenface space's dimension becomes larger (>10) recognition rate increases.

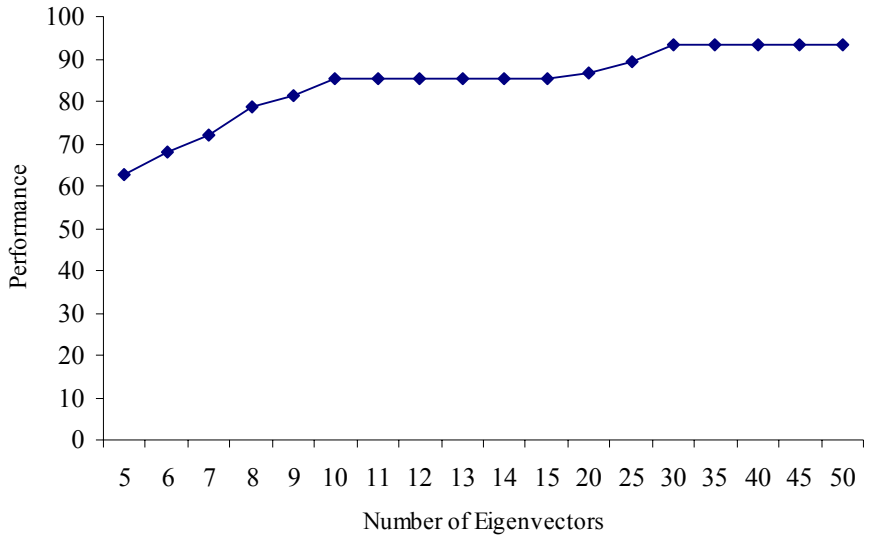


Figure 4.19: Performance With Respect to Number of Eigenvectors in (Subspace LDA- METU Vision Face Database)

The experiment due to change in lighting conditions is done by changing the brightness of all testing images by –20%, -10%, 0%, +10% and +20%.

Performance of Subpace LDA is better to recognize images with brightness variation when compared to Eigenface methods. Brightness variation

does not affect the Subpace LDA method as largely as the performance of Eigenface, so Subpace LDA is more stable than Eigenface when brightness varies greatly. Recognition performance of this method against illumination changes can be observed in Table 4.10 (a), (b), (c), (d),(e), and (f) below. It can be seen from results that histogram equalization decreases the performance when brightness variation is less. If there is much change in brightness, using histogram equalization as a preprocessing step increases the performance so it makes the algorithm more robust.

Table 4.10: Performance due to variation in brightness

(a) (SubpaceLDA)- # of Eigenvectors: 15

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	44%	81.33%	85.33%	82.66%	50.66%
Histogram Equalized	77.33%	78.66%	78.66%	78.66%	78.66%

(b) (SubpaceLDA)- # of Eigenvectors: 20

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	62.66%	78.66%	85.3%	84.00%	78.66%
Histogram Equalized	78.66%	81.33%	81.33%	81.33%	81.33%

(c) (SubpaceLDA)- # of Eigenvectors: 25

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	70.6%	88%	89.3%	88%	74.66%
Histogram Equalized	77.33%	78.66%	78.66%	78.66%	77.33%

(d) (SubpaceLDA)- # of Eigenvectors: 30

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	65.3%	84.0%	93.3%	86.6%	74.66%
Histogram Equalized	82.66%	82.66%	82.66%	81.33%	82.66%

(e) (SubpaceLDA)- # of Eigenvectors: 35

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	66.6%	82.6%	93.3%	80%	76.0%
Histogram Equalized	80.0%	82.66%	84.0%	82.66%	84.0%

(f) (SubpaceLDA)- # of Eigenvectors: 40

Brightness	-20%	-10%	0%	+10%	+20%
Recognition rate	56%	81.33%	93.3%	85.33%	64%
Histogram Equalized	82.66%	82.66%	82.66%	82.66%	81.33%

4.3 Performance Comparison

Table 4.11 shows the test results of all the methods implemented in this thesis in different testing conditions. It should be noted that these are the best values for the test cases below.

Table 4.11: Performance Comparison Table

Approach	Face Database	Test Input Property	Recognition Rate
Eigenface	Yale	No glasses images	80.0%
Eigenface	Yale	Normal lighting & no glasses images	82.0%
Eigenface	Yale	Variation in lighting & expression	81.0%
Eigenface	AT&T	1 training input Variation in Expression	70.0%
Eigenface	AT&T	5 training Inputs Variation in Expression	82.0%
Eigenface	AT&T	9 training Inputs Variation In Expression	80.0%
Eigenface	METU Vision	Lighting Variation: 0%	80.0%
Eigenface	METU Vision	Lighting Variation:10%	66.66%
Eigenface + Hist Eq.	METU Vision	Lighting Variation:10%	69.33%
Eigenface	METU Vision	Lighting Variation:20%	20.0%
Eigenface+ Hist Eq.	METU Vision	Lighting Variation:20%	70.66%
Kernel Eigenface	Yale	No glasses images in training set	81.82%
Kernel Eigenface	Yale	Normal lighting & no glasses images	85.0%
Kernel Eigenface	Yale	Variation in lighting & expression	84.55%
Kernel Eigenface	AT&T	5 training Inputs Variation in Expression	85.0%
Kernel Eigenface	METU Vision	Lighting Variation: 0%	84.0%
Kernel Eigenface	METU Vision	Lighting Variation:10%	70.6%
Kernel Eigenface + Hist Eq.	METU Vision	Lighting Variation:10%	73.33%
Kernel Eigenface	METU Vision	Lighting Variation:20%	40.6%
Kernel Eigenface + Hist Eq.	METU Vision	Lighting Variation:20%	73.33%
Subpace LDA	Yale	No glasses images in training set	91.3%

Table 4.11: Performance Comparison Table (cont'd)

Subpace LDA	Yale	Normal lighting & no glasses images	85.5%
Subpace LDA	Yale	Variation in lighting & expression	90.3%
Subpace LDA	AT&T	1 training input Variation in Expression	74.5%
Subpace LDA	AT&T	5 training Inputs Variation in Expression	85.0%
Subpace LDA	AT&T	9 training Inputs Variation In Expression	83.0%
Subpace LDA	METU Vision	Lighting Variation: 0%	93.3%
Subpace LDA	METU Vision	Lighting Variation:10%	80.0%
Subpace LDA+ Hist Eq.	METU Vision	Lighting Variation:10%	88.0%
Subpace LDA	METU Vision	Lighting Variation:20%	74.6%
Subpace LDA+ Hist Eq.	METU Vision	Lighting Variation:20%	82.66%

CHAPTER 5

CONCLUSIONS

In order to find the best values for the parameters in the training stage, a number of experiments are performed for Principal Component Analysis (PCA), Kernel PCA and Subspace Linear Discriminant Analysis (Subspace LDA) and their performances are evaluated.

The investigated techniques are previously proposed and studied in the literature. PCA was proposed in 1991, and based on PCA, Kernel PCA and Subspace LDA were proposed later. They were also tested under different sets of face databases. The main purpose is studying the performance of Subspace LDA and Kernel PCA methods in depth, which are two competeting methods reported to be among the best performing methods in the literature.

The Eigenface approach to face recognition was motivated by information theory, leading to the idea of basing face recognition on a small set of image features that best approximate the set of known face images, without requiring that they correspond to our intuitive notions of facial parts and features. Eigenface uses principal components analysis (PCA) for dimensionality reduction. The main idea is to find the principal components of the distribution of faces considering the eigenvectors of the covariance matrix of the set of face images. Each image is represented as a linear combination of eigenvectors. With this simple transformation, it reduces a (image size M) dimensional problem to a

(number of image M') dimensional problem, where M is the image size while M' is the number of face images.

The Eigenface approach has advantages in its speed and simplicity, learning capacity, and relative insensitivity to small or gradual changes in facial expression of face images. However, brightness variation severely affects the performance. Eigenface only calculates the total scatter matrix. If the training set contains images of faces captured under varying illumination, the projection matrix will contain principal components due to variation in lighting. Then, the points in the projected space will not be well clustered, and worse, the classes may be smeared together.

Kernel Eigenface method is more robust and with respect to Eigenface method. Much of the important informaton may be contained in the high order relationships among the pixels of a face image, so non-linear features such as, brightness and facial expressions can be extracted by using Kernel PCA for face recognition. In this thesis, Gaussian Radial Basis Function kernel [46] is not used for higher-order transformation, so the performance is not as expected.

LDA tries to avoid clustering problem in Eigenface method by finding the projection matrix that maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples.

LDA is better than Eigenface to recognize images with modified facial features (i.e. mustaches, glasses, etc.) and varying illumination. However, the

speed of LDA is much slower than that of Eigenface as more scatter covariance matrices have to be computed. The other shortcoming of LDA is that it cannot be applied to face recognition problems where only one training image per person are available, such as suspect identification.

Both Eigenface and LDA have merits and drawbacks. Before choosing one of these approaches for face recognition, various aspects have to be compared. Application using face recognition can be developed with Eigenface or LDA approach.

To develop a real application, detailed design has to be done. More problems have to be solved. A colour image needs to be converted to a grayscale image. Masking is required to mask the background region of the image to reduce the error rate due to distractions from the background.

Compared with Eigenface, which extracts Most Expressive Features, LDA is designed to extract features more suitable for classification purpose, so called Most Discriminating Features.

Eigenface finds the projection matrix that maximizes the total scatter across all images, while LDA finds the projection matrix that maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples.

In idealized condition, with only different facial expression and no light illumination changes, both Eigenface and LDA perform well. On the other hand, Eigenface has difficulty in recognizing faces with modified facial features (i.e.

mustaches, glasses, etc.) and varying illumination. Although, removing the first three principal components does improve the performance of Eigenface in the presence of illumination variation.

LDA, however, is much better on recognizing faces with modified facial features and varying illumination. It is because LDA tends to discount those portions of the image that are not significant for recognizing an individual.

Compared with Eigenface, LDA is slower. For Eigenface, only the total scatter matrix has to be computed for finding the eigenvectors to form "face space". In LDA, the within-class scatter matrix, between-class scatter matrix and the generalized eigenvectors have to be found. Also, to avoid the within-class matrix be singular, PCA has to be done to reduce the with-class scatter matrix. All these processes are time consuming and becomes problematic if the training set is large.

One solution is to compute all the matrix operations in parallel, which speeds it up greatly.

In Eigenface, the shape and location of the original data sets changes when transformed to a different space whereas LDA doesn't change the location but only tries to provide more class separability and puts a decision region between the given classes. LDA provides better understanding on the distribution of the feature data more understandable.

For both Eigenface and LDA, recognition is efficient only when the number of classes is larger than the dimensions of the "face space"; otherwise,

the projection of an unknown image requires pixel-by-pixel multiplication of the input image by all eigenfaces/fisherfaces. This multiplication is equivalent to or worse than template-matching with respect to computation time since an extra distance calculation is needed in the subspace. However, the occurrence of class overlapping increases when more face classes are represented by the same face space, thus lowering the recognition rate.

For both Eigenface and LDA, when the training set is updated by adding or removing a person or a class in the training set, the "face space" has to be formed again. Therefore, it is not suitable for those applications which need to update the face database frequently.

For Eigenface, it needs one photo for each person for training. But, for LDA, it requires several images for training, which are not always available for some applications like suspect recognition. One solution is to derive multiple samples from a single face image, by which LDA can be trained.

REFERENCES

[1] R.A. Hutchinson and W.J. Welsh. Comparison of neural networks and conventional techniques for feature location in facial images. *IEE International Conference on Arti_cial Neural Networks*, Conf. Publication Number 313:201-205, 1989.

[2] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 15, October 1993.

[3] A. Pentland, B. Moghaddam, and T. Starner, "View based and modular eigenspaces for face recognition," in *Proceedings on IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 84-91, 1994.

[4] A.L. Yuille, P.W. Hallinan, and D.S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99-111, 1992.

[5] M. Kass, A. Witkin, and D. Terzopoulos. SNAKES: Active Contour Models. *Proceedings of the International Conference on Computer Vision*, pages 259-268, 1987.

[6] R.J. Baron. Mechanisms of human facial recognition. *International Journal on Man- Machine Studies*, 15:137-178, 1981.

[7] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042-1052, 1993.

[8] O. Nakamura, S. Mathur, and T. Minami. Identification of human faces based on isodensity maps. *Pattern Recognition*, 24(3):263-272, 1991.

[9] K. Sutherland, D. Renshaw, and P.B. Denyer. A novel automatic face recognition algorithm employing vector quantisation. *IEE Colloquium on \Machine Storage and Recognition of Faces*", 4:1-4(Digest No: 1992/017), 1992.

[10] C.S. Ramsay, K. Sutherland, D. Renshaw, and P.B. Denyer. A comparison of vector quantization codebook generation algorithms applied to face recognition. In *British Machine Vision Conference*, pages 508-517. Springer-Verlag, 1992.

[11] N.M. Allinson, A.W. Ellis, B.M. Flude, and A.J. Luckman. A connectionist model of familiar face recognition. *IEE Colloquium on \Machine Storage and Recognition of Faces"*, 5:1-10(Digest No: 1992/017), 1992.

[12] T. Kohonen. Springer-Verlag *Self-organization and associative memory*, 2nd edition, 1988.

[13] G.W. Cottrell and M. Fleming. Face recognition using unsupervised feature extraction. *International Neural Network Conference*, 1:322-325, 1990.

[14] T.J. Stonham. Practical face recognition and veri_cation with WISARD. In H.D. Ellis, M.A. Jeeves, F. Newcombe, and A. Young, editors, *Aspects of face processing* pages 426-441. Martinus Nijho_ Publishers, 1986.

[15] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Training models of shape from sets of examples. In *British Machine Vision Conference*, pages 9{18. Springer-Verlag, 1992.

[16] L. Gillick and S.J. Cox. Some statistical issues in the comparison of speech recognition algorithms. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 532{535, 1989.

[17] D.E. Goldberg. *Genetic algorithms in search, optimization, and machine learning,* Addison-Wesley, 1989.

[18] D. Beymer, "Face recognition under varying pose," in *Proceedings of 23rd Image Understanding Workshop*, vol. 2, pp. 837-842, 1994.

[19] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proceedings of International Conference on Pattern Recognition*, pp. 586-591, 1991.

[20] A.J. Goldstein, L.D. Harmon, and A.B. Lesk. *Identification of human faces. Proceedings of the IEEE*, 59(5):748{760, 1971.

[21] R.Epstien, P. Hallina, and A.Yuille, "5+/-eigenimages suffices: An empirical investigation of low-dimensional ligting models," in *Proceedings of the Work-shop on Physics-based Modeling in Computer Vision*, pp.108-116, 1995.

[22] H.Murase and S.Nayar, "Visual learning and recognition of 3-d objects from appearance," *International Journal of Computer Vision*, vol.14, pp.5-24,1995

[23] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of IEEE*, vol. 77, pp. 257-286, February 1989.

[24] L.Rabiner and B. Huang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[25] E. Levin and R. Pieraccini, "Dynamic planar warping for optical character recognition,"in *Internatonal Conference on Acoustics, Speech and Signal Processing*, pp. 149-152, 1992.

[26] O. Agazzi, S. Kuo, E.Levin, and R. Pieraccini,"Connected ad degraded text recognition using p;anar HMM," in *International Conference on Acoustics, Speech ad Signal Processing* '93, vol. 5, pp.113-116, 1993

[27] S. Akamatsu, T. Sasaki, H. Fukumachi, and Y. Suenuga, "A robust face identification scheme-KL expansion of an invariant feature space," *in SPIE Proceedings::Intelligent Robots and Computer Vision X:Algorithms and Techniques*, vol. 1607, pp. 71-84, 1991.

[28] F. Samaria, "Face segmentation for identification using hidden Markov models" in *British Machine Vision Conference*, 1993.

[29] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs Fisherfaces:Recognition using class specific linear projection," in *Proceedings of Fourth European Conference on Computer Vision, ECCV'96*, pp. 45-58, April 1996.

[30] A. W. Senior, "Face and feature finding for a face recognition system," in *Second International Conference on Audio and Video-Based Biometric Person Authentication*, pp. 154-159, 1999.

[31] F. Samaria and F. Fallside, "Face identification and feature extraction using hidden Markov models," *Image Processing: Theory and Applications*, 1993.

[32] F. Samaria and F. Fallside, "Automated face identification using hidden Markov models," in Proceedings of the *International Conference on Adavanced Mechatronics,* pp. 1-9, 1993.

[33] F. Samaria and A. Harter, "Parametrisation of stochastic model for human face identification," in *Proceedings of the second IEEE Workshop on Application of Computer Vision,* pp. 138-142, 1994.

[34] F. Samaria, *Face Recognition Using Hidden Markov Models*. PhD thesis, U niversity of Cambridge, 1994.

[35] Y. Cheng, K. Liu, J. Yang, Y. Zhang, and N. Gu, "Human face recognition method based on the statistical model of small sample size," in *SPIE Proceedings:Intelligent Robots and Computer Vision X:Alg. And Techn*, vol. 1607, pp. 85-95, 1991.

[36] F. Samaria and S. Young, "HMM based architecture for face identification," Image and Vision Computing, vol. 12, pp. 537-

[37] Y. Cheng, K. Liu, J. Yang, and H. Wang, "A robust algebraic method for human face recognition," in *Proceedings of 11th International Conference on Pattern Recognition*, pp. 221-224, 1992.

[38] B. Moghaddam and A. Pentland, Probabilistic visual learning for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 19(7):696 710, 1997.

[39] H. Murase and S. Nayar. Visual leraning and recognition of 3d objects from appearance. *International Journal of Comupter Vision*, 14(1):5-24, 1995

[40] M. J. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63-84, 1996.

[41] A. Rajagopalan, K. Kumar, J. Karlekar, R. Manivasakan, M. Patil, U. Desai, P. Poonacha, and S. Chaudhuri. Finding faces in photographs. In *Proceedings of the Sixth International Conference on Copmputer Vision*, pages 640-645, 1998.

[42] G. Cotrell and M. Fleming, "Face recognition using unsupervised feature extraction," in *Proceedings International neural Network Conference*, pp. 322-325, 1990.

[43] A. Lawrence, C. Giles, A. Tsoi, and A. Back, "Face recognition: A convolutional neural network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98-113, 1997.

[44] A. N. Rajagopalan, P. Burlina, and R. Chellappa Higher order statistical learning for vehicle detection in images. In *Proceedings of the Seventh International Conference on Computer Vision,* volume 2, pages 1204 1209, 1999.

[45] S.-H. Lin, S.-Y. Kung, and L.-J. Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Transactions on Neural Network,* vol. 8, pp. 114-132, January 1997.

[46] B. E. Boser, 1. M. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifthe Annual Workshop on Computational Learning Theory*, pages 144-152, 1992.

[47] B. Schölkopf, A. Smola, and K-R. Müller. Non-linear component analysis asa kernel eigenvalue problem. *Neural Computation*, 10(5):1299-1319, 1998.

[48] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. Malsburg, and R. Wurtz, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Transactions on Computers*, vol. 42. no. 3, pp. 300-311, 1993.

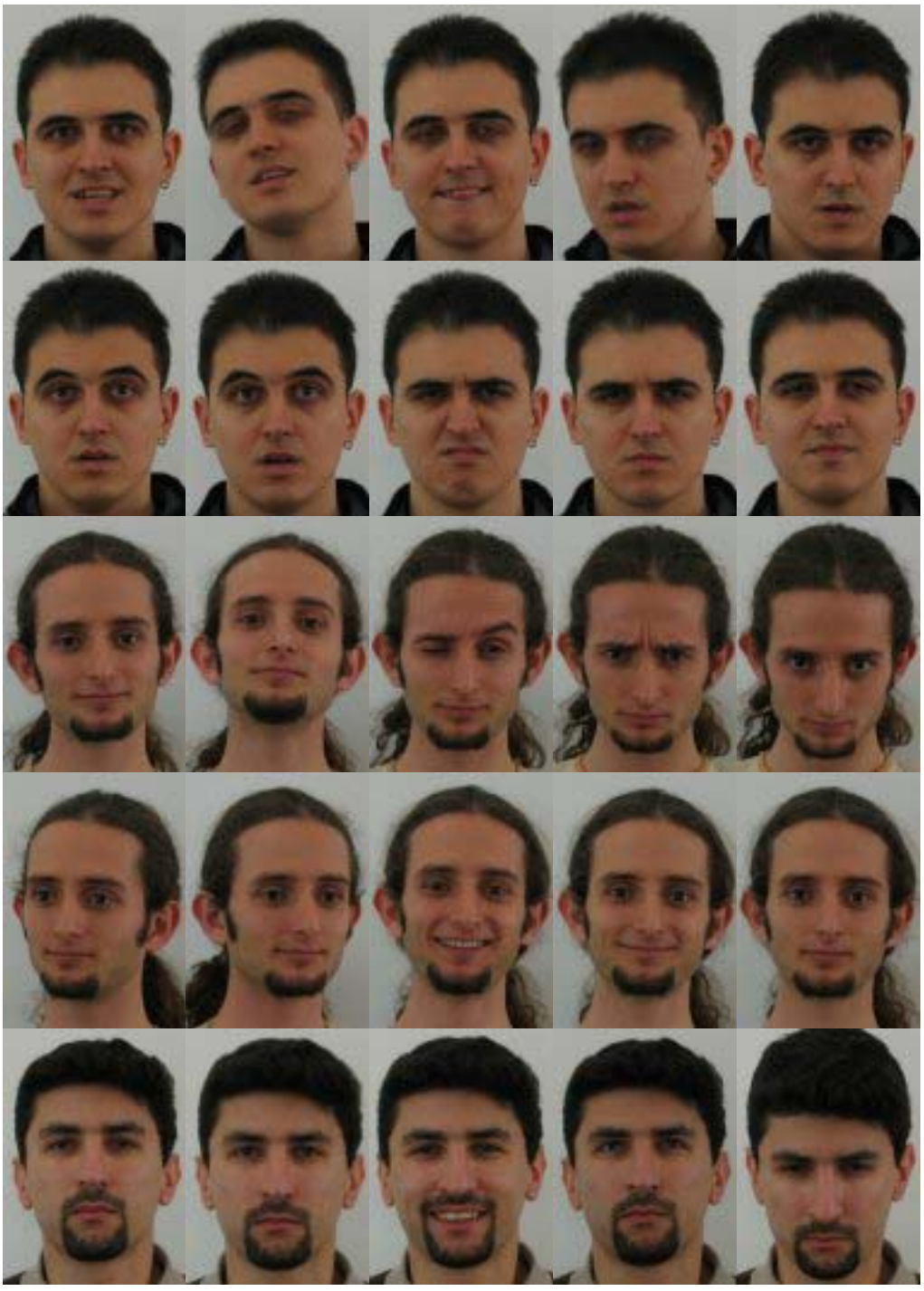
[49] J. Ben-Arie and D. Nandy, "A volumetric/iconic frequency domain representation for objects with application for pose invariant face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 449-457, may 1998.

[50] J. Zhang, Y. Yan, and M. Lades, "Face recognition: Eigenface, elastic matching and neural nets," *Proceedings of the IEEE*, vol. 85, pp. 1423-1435, September 1997.

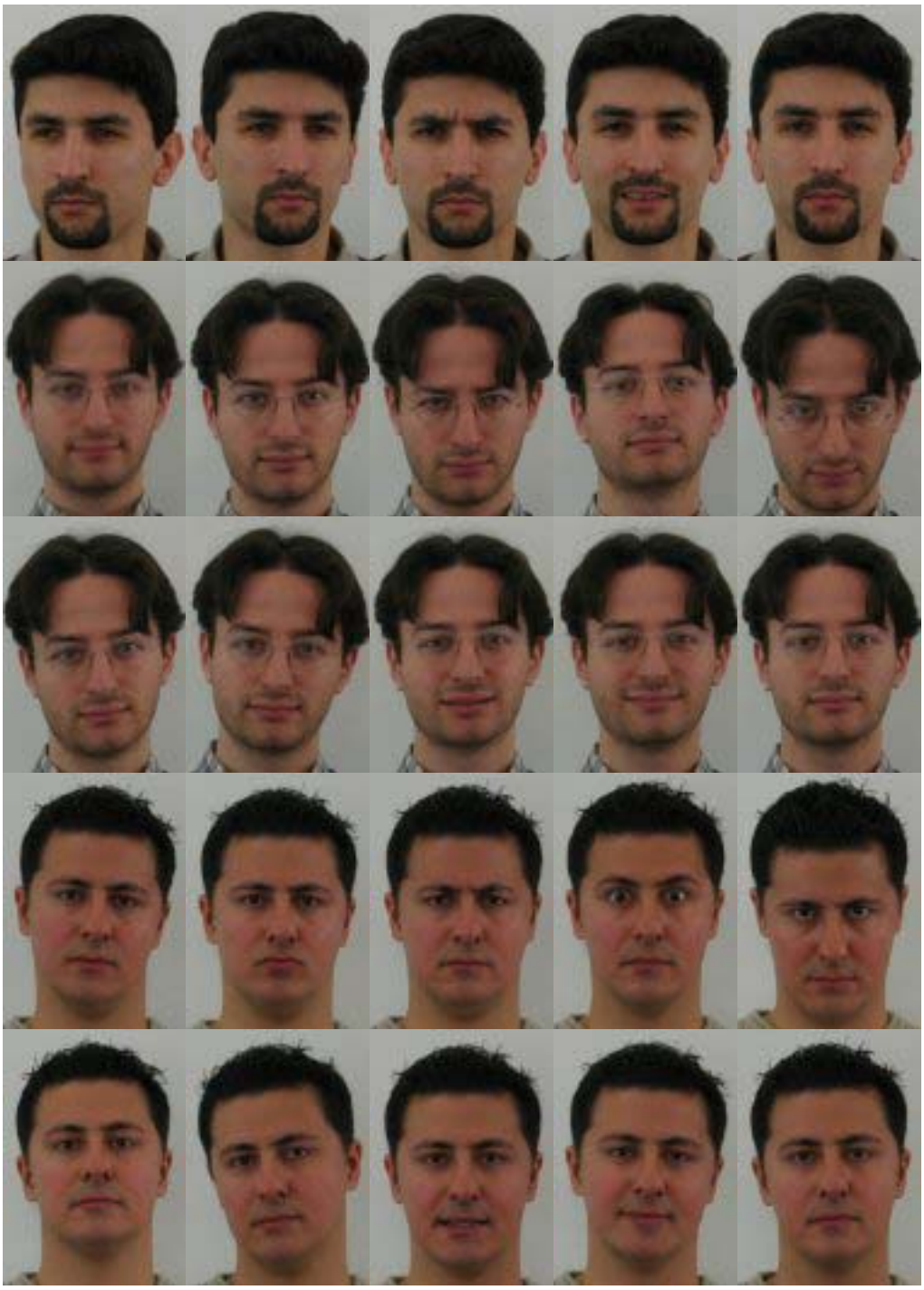
APPENDIX

APPENDIX A

METU VISON LAB FACE DATABASE

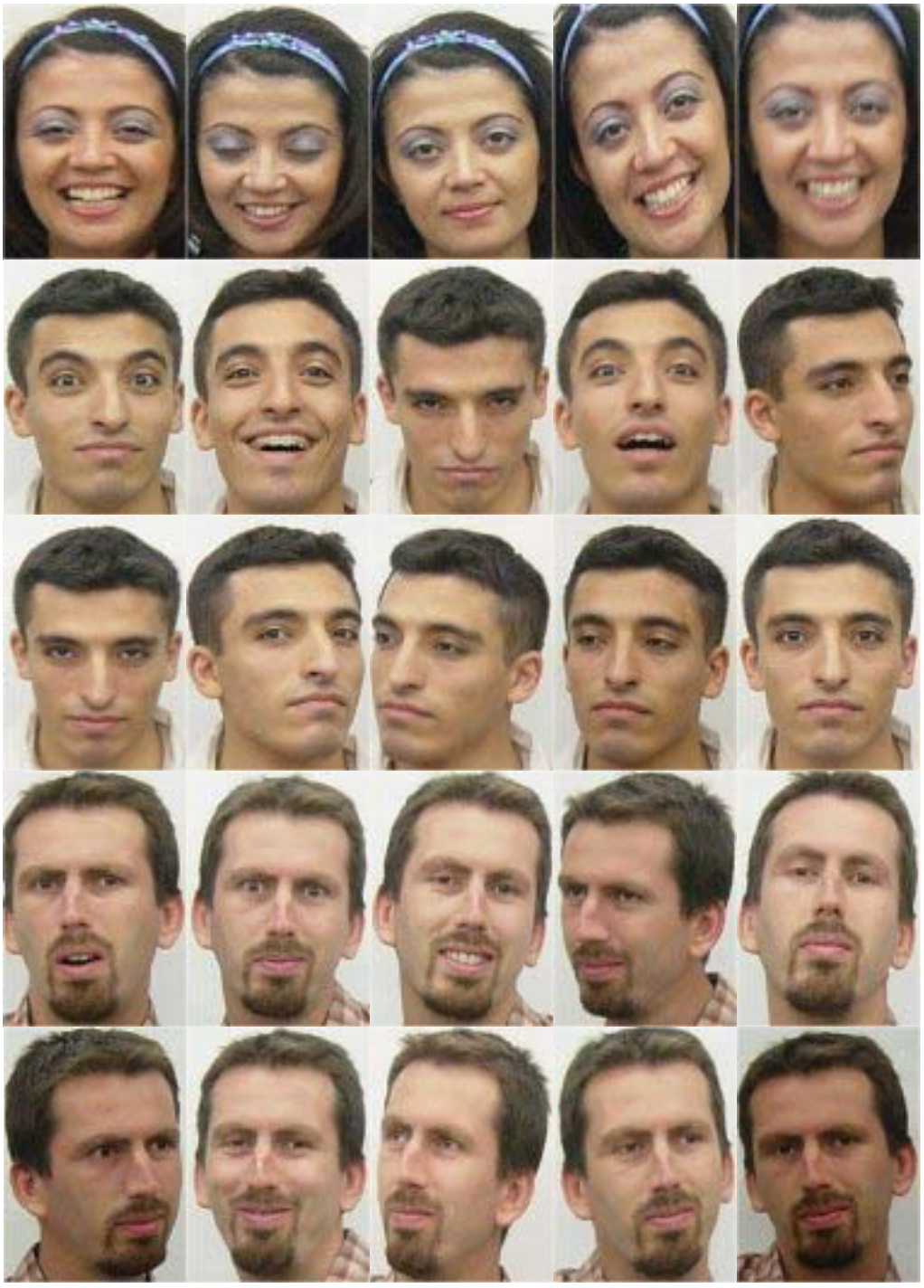


METU VISON LAB FACE DATABASE (CONT'D)



METU VISON LAB FACE DATABASE (CONT'D)

METU VISON LAB FACE DATABASE (CONT'D)

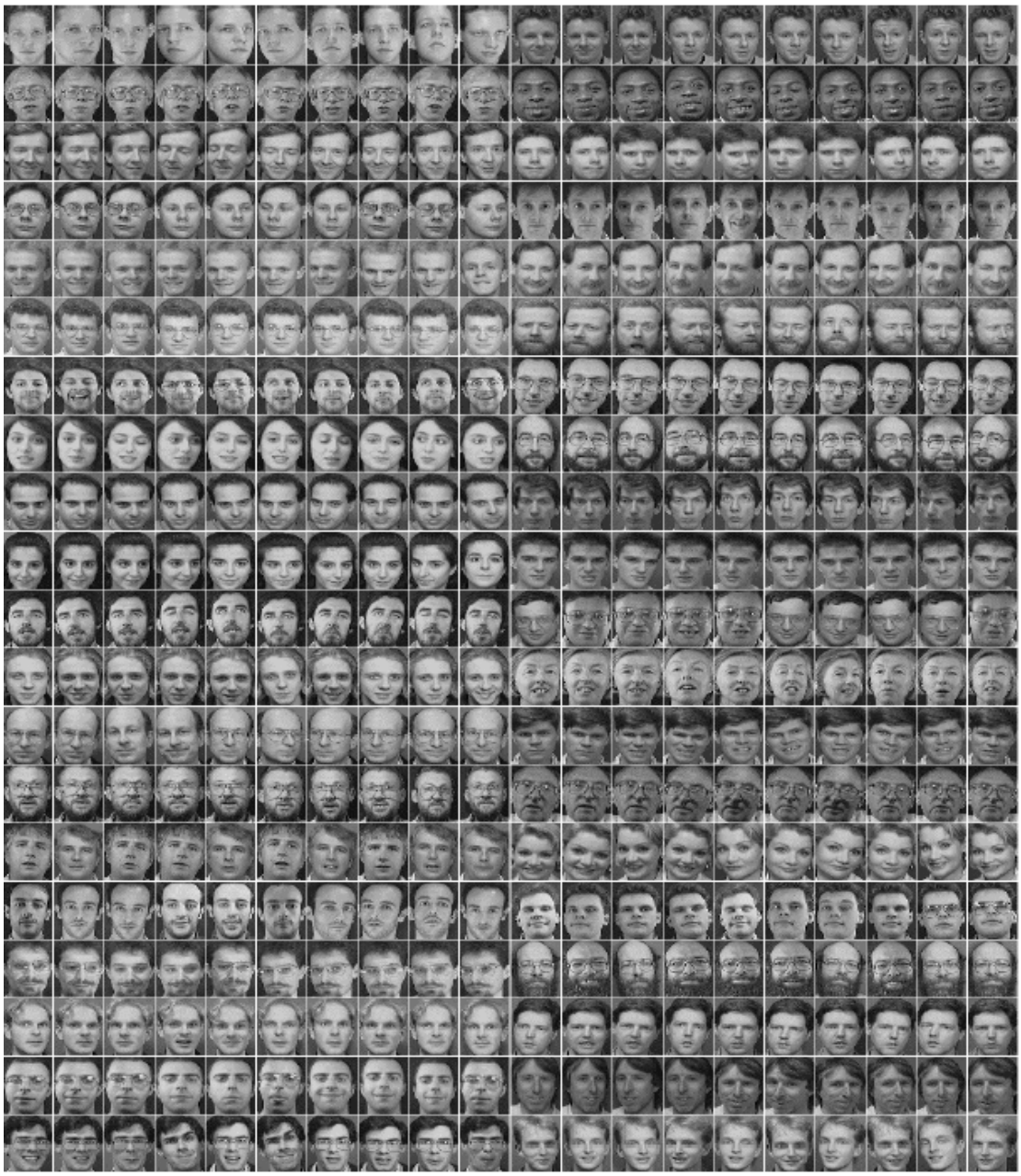


METU VISON LAB FACE DATABASE (CONT'D)

METU VISON LAB FACE DATABASE (CONT'D)

APPENDIX B

AT&T FACE DATABASE

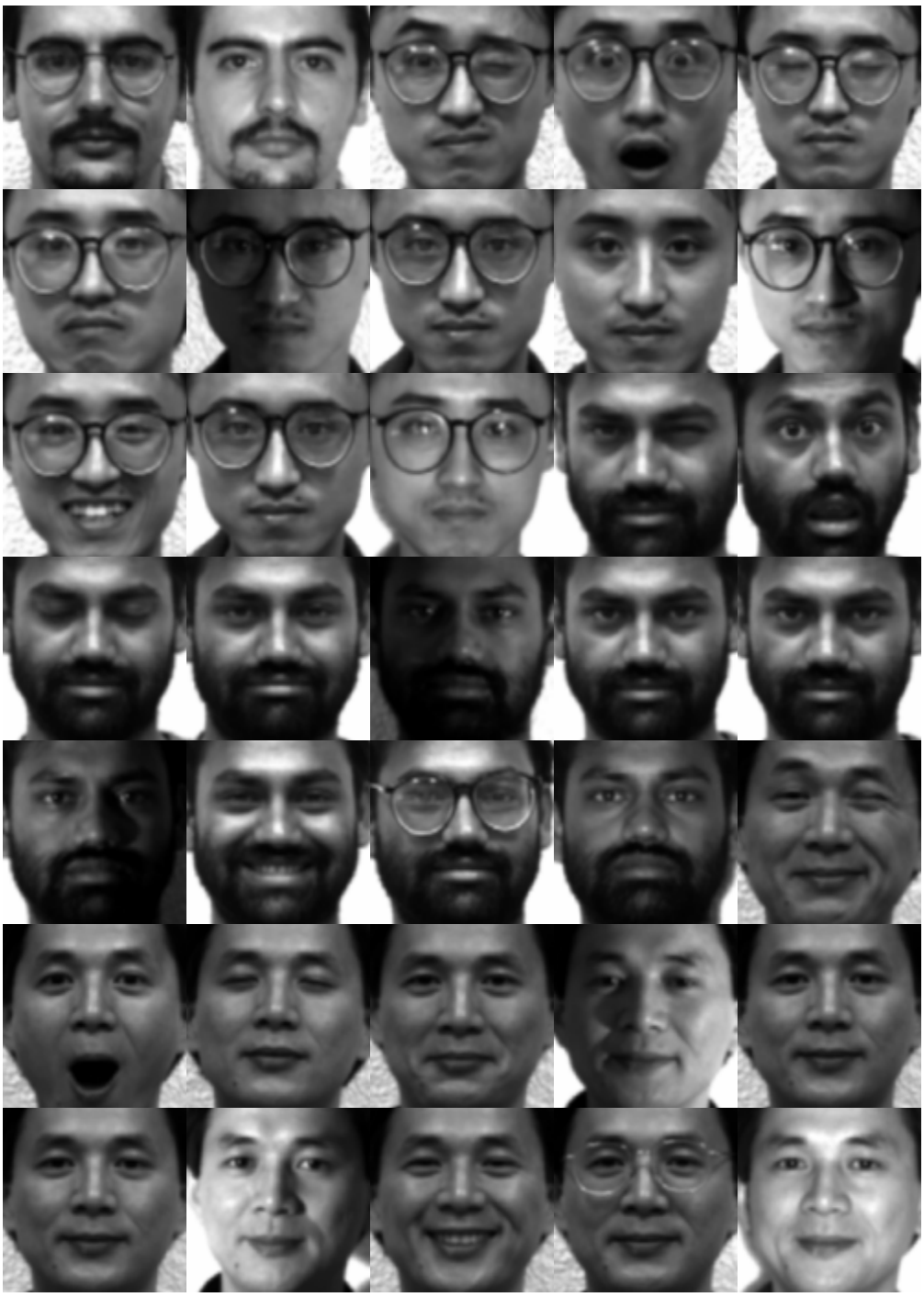


APPENDIX C

YALE DATABASE

YALE DATABASE (CONT'D)

YALE DATABASE (CONT'D)



YALE DATABASE (CONT'D)

YALE DATABASE (CONT'D)