Comparative Analysis of Face Recognition Algorithms and Investigation on the Significance of Color

Behnam Karimi

A Thesis

In

The Department

of

Computer Science and Software Engineering

Presented in Partial fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montreal, Quebec, Canada

August 2006

Your file Votre référence
ISBN: 978-0-494-30081-7
Our file Notre référence
ISBN: 978-0-494-30081-7

ABSTRACT

Comparative Analysis of Face Recognition Algorithms and Investigation on the Significance of Color

Behnam Karimi

Face recognition technology has rapidly evolved and become more popular in recent years. It is being used for both research and other applications such as security systems. One of the key challenges in face recognition systems is to identify the role of different cues for face identification. The role of color, which appears to be a salient attribute of faces, is debated within the literature. Some research has suggested that it confers little recognition advantage for identifying faces. Other research suggests that color is a very important cue in face identification. The clear perception of colors in the environment illustrates that color must be important in the interpretation of complex scenes and recognizing objects in the environment.

In this thesis several face recognition methods are introduced. Experiments were conducted using both grayscale and color images. The accuracy of each algorithm has been identified and a comparison was performed between them in terms of recognition rates. A system is also proposed, which uses color features for face recognition. This system can be used by different face recognition algorithms. The goal of this study is to show the differences between some popular face recognition methods (new and traditional) and also the role of color in face recognition using different face recognition algorithms. The results of our experiments show that using color improves the recognition rate for traditional and new methods. The improvement is more obvious for traditional methods because the recognition rate is not near the peak.

Acknowledgement

Special thank to Professor Adam Krzyzak, my supervisor, who helped me to complete this work and was available whenever I needed help.

Also I have to thank Mr. Peter Peer in University of Ljubljana (Slovenia), Faculty of Computer and Information Science, Computer Vision Laboratory, who provided me CVL Face Database, which was used in this research.

Dedication

This is dedicated to my son, whose smile gave me hope to complete this work and to my wife who gave me support and encouragement.

Table of Contents

List of Tables (Figures)

1. Introduction

Face recognition is considered to be a part of pattern recognition technology. Recognition and specifically face recognition covers a range of activities from many aspects of life. .Face recognition and recognition of moving people in natural scenes requires that a set of visual activities be performed. This process consists of three tasks: acquisition, normalization and recognition. Acquisition means the detection and tracking of face-like image patches in dynamic scenes. It is followed by line and alignement normalization.. And finally recognition is the representation and modeling of face images as identities, and the association of novel face images with known models.

A number of approaches for face recognition and classification have been proposed in the literature. These can be termed as Principal Components Analysis (PCA) [33], Fisher's Linear Discriminant (FLD) [35], Discriminative Common Vectors [13] and Laplacianfaces [52].

Techniques like PCA and FLD treat the face image as a vector in a high-dimensional space and derive a lower dimensional representation (in the case of PCA) or a discriminatory representation (in the case of FLD). FLD provides better performance but is computationally more intensive compared to feature-based approaches. Also, the performance of data analysis techniques depends on the training data. Discriminative Common Vectors approach provides better performance because a common vector is chosen for each class in the dataset instead of dealing with several faces in each class. In the Laplacianfaces approach, different from PCA and FLD, preserving local information is being focused in order to detect the essential manifold structure.

Most of the recognition research is performed on gray-scaled datasets. Color is not taken into consideration in the majority of these studies. This thesis will compare recognition rates between different traditional and new face recognition methods, and attempt to identify the role of color in face recognition systems. This thesis also argues that the role of color in face recognition could have a significant impact on recognition performance and should be a factor in face recognition technologies [3].

1.1 Applications of face recognition

The applications of Face Recognition Techniques (FRT) can be divided into commercial and law enforcement applications. A list detailing the applications along with their constraints is shown in table 1.1.

	Applications	Advantages	Disadvantages
1a.	Credit card, driver's license, passport, and personal identification	Controlled image Controlled segmentation Good quality images	No public databases Large potential database
1b.	Mug shots matching	Mixed image quality More than one image available	No public databases Large potential database Rare search type
2.	Bank / store security	High value Geographically localized search	Uncontrolled segmentation Low image quality
3.	Crowd surveillance	High value Small file size Availability of video images	Uncontrolled segmentation Low image quality Real-time
4.	Expert identification	High value Enhancement possible	Low image quality Legal certainty required
5.	Witness face reconstruction	Witness search limits	Unknown similarity
6.	Electronic mug shots book	Descriptor search limits	Viewer fatigue
7.	Electronic lineup	Descriptor search limits	Viewer fatigue

Table 1.1 - Applications and their constraints.

These applications can be classified into two groups: Still images and real-time dynamic images. Among the groups, significant differences exist, depending on the specific application. Differences include variations in image quality, amount of background clutter, and the nature, type and amount of input from a human (as in application 4 and 5).

Applications 1, 2, and 3 involve matching one face image to another face image. Applications 4-7 involve finding or creating a face image, which is similar to the human recollection of a face. Each of these applications imposes different requirements on the recognition process. Matching requires that the candidate matching face image be in a set of face images selected by the system. Similarity detection requires, in addition to matching, that images of faces be found which are similar to a recalled face; this requires that the similarity measure used by the recognition system closely match the similarity measures used by humans.

1.2 Stages in Face Recognition

The definition of the task in the face recognition process is: to identify one or more persons in a scene using a stored database of faces. The solution of the general problem is divided into three different stages:

- segmentation of faces from cluttered scenes,
- extraction of features from the face region,
- decision

Segmentation is usually achieved by the following algorithm: An edge map is created, and then the edges are connected together using several heuristics and the edges are matched into an elliptical shape using a Hough transformation.

If the input is composed of video images (moving objects), motion could be used for segmentation.

The second and most important stage is the extraction of the features. There are two types of features: holistic features (where each feature is a characteristic of the whole face) and partial features (nose, hair, mouth eyes, etc.). Partial features techniques use crucial points in the face for recognition, whereas holistic feature techniques always consider the face as a whole. For example PCA is a holistic feature technique.

In the third and last stage a decision is made using the data collected from the previous stage. There will be three types of decisions that can be made depending on the application: 1) Identification, where labels for each individual must be obtained; 2) recognition of a person, where a decision is made based on the face that the individual has already been seen, and; 3) categorization, in which the face must be assigned to a certain category. This thesis will focus on two stages of face recognition: extraction of features and decision.

2. Why Face Recognition?

When there is a requirement for determining people's identity, the obvious question is what technology is best at supplying this information? There are many ways that humans can identify each other, and it is the same for machines. There are many different identification methods available, many of which have been used in commercial applications for years. The most common type of application is Password/PIN known as Personal Identification Number systems. The problem with this or other similar methods is that they are not unique, and it is possible for a person to forget, loose or even have it stolen by someone else. To solve these problems there has been considerable interest in "biometrics" identification systems, which use pattern recognition methods to identify people using characteristic features. However, these techniques are not easy to use and can be intrusive both physically and socially. For example, in bank transactions and entry into secure areas, the user must position the body relative to the sensor, and then pause for a second to declare him or herself since the quality of the image is important [30].

While the pause and present interaction is useful in high-security situations, it's inefficient in other circumstances. For instance, a store that wishes to recognize its best customers, or an information kiosk that remembers you, or a house that knows the people who live there does not want to interrupt the individual's daily activities. Face recognition from video and voice recognition have a natural place in these next generation smart environments, they are unobtrusive, are usually passive, do not restrict user movement, and are now both low power and inexpensive. Perhaps most important, however, is that humans identify other people by their face and voice, and are likely to be comfortable with systems that use face and voice recognition [30].

2.1 History of Face Recognition

Face recognition was established when machines started to become more intelligent and were able to fill in, correct or help the lack of human abilities and senses. Face recognition and computer vision subjects are both important because of the practical importance of the topic and theoretical interest from cognitive science. Face recognition is not the only method of recognizing images and people. Today machines are used for different recognition purposes such as fingerprinting or iris scans. These methods of identification are more accurate than face recognition but face recognition is more interesting for researchers because of its non-invasive nature and because it is the primary method used by humans for identifying people.

From the beginning there have been two main approaches in face recognition technologies: 1) geometrical approach and; 2) pictorial approach. The geometrical approach uses the spatial configuration of facial features. That means the main geometrical features of the face such as the eyes, nose and mouth are first located and then faces are grouped or classified on the basis of various geometrical distances and angles between features. The pictorial approach uses templates of the facial features. It uses templates of the major facial features and entire face to perform recognition on frontal views of faces. Many of the projects that were based on these two approaches have some common extensions that handle different pose backgrounds. Apart from these two techniques there are other recent template-based approaches, which form templates from the image gradient, and the principal component analysis approach, which can be read as a sub-optimal template approach. Finally there is the deformable template approach that combines elements of both the pictorial and feature geometry approaches and has been applied to faces in varying poses and expressions [12, 30].

Since the early start of face recognition there has been a strong relationship with the science of neural networks. The earliest and best known example of a face recognition "system", using neural networks, is the Kohonen model. That system was a simple neural network that was able to perform face recognition for aligned and normalized face images. The type of network employed computed a face description by approximating the eigenvectors of the face image's auto-correlation matrix; these eigenvectors are now known as "eigenfaces" [12, 27].

Following the Kohonen model other methods were developed based on older techniques. To summarize the methods, the "idea" of face recognition was first based on a geometrical approach or pictorial approach, and after that methods like eigenfaces, Principal Component Analysis, or other methods that process images in combination with neural networks or other expert systems were developed.

2.2 The Present

With the rapid evolution of the technology and the commercialization of technological achievements, face recognition became more and more popular, not only for research but also for use in security systems. This provided incentive for many researchers and companies to develop methods that would automatically recognize faces that could be used for many applications, including security and human-computer interaction. For example, a face recognition machine could be developed to allow automated access to buildings or enable a computer to recognize the person sitting at the console. Most of the face recognition systems work only for frontal or near frontal images. This is a challenge in face recognition to make recognition systems less rigid.

2.3 Face Recognition and Face Detection

As discussed earlier, face recognition is a technique that is used for recognizing faces but it is not necessary to “freeze” the user to take a picture. The problem with recognizing a face arises when the pose of the face is different, but in particular, there is a limit on face rotations in depth, which include left and right and up and down rotations. Face recognition is a difficult task because it has to discriminate among similar objects. That means when two faces are similar, recognition is going to be a challenge. By adding pose to a face, the problem becomes more complex. The appearance of face changes under rotation since the face has a complex three dimensional structure.

Face recognition and face detection are two distinct processes. The main difference is that face recognition is a technique to detect faces and search through a dataset in order to find an exact match but on the other hand face detection is looking for any match and as soon as a match is found the search stops.

In face recognition techniques, three visual cues are most important: motion, color and face appearance. The visual cue of color is important but most of the implementations of face recognition algorithms do not use color databases. This paper focuses upon person identification within such a framework. That means complementary to face recognition approach, color is also considered as an important cue and a system is proposed to measure the importance of color in face recognition.

In general most of research efforts have been concentrated on face recognition tasks in which only a single image or a few images of each person are available. The scalability of the recognition systems has been a major concern in large databases, which contain thousands of people. Face recognition tasks require recognition to be performed using sequences acquired and normalized automatically in poorly constrained dynamic scenes. These are characterized by low resolution, large-scale changes, variable illumination and occasionally inaccurate cropping and alignment. Recognition based upon isolated images of this kind is highly inconsistent and unreliable. However, accumulating recognition scores over time can compensate the poor quality of the data.

Face recognition is an area of research, which involves different fields such as physics, biology, psychology, mathematics, and computer science. Different face recognition methods have been proposed in recent years resulting in interesting applications. The research presented here looks at one of the core problems in a face identification system, which is the role of color in face recognition systems.

3. Face Recognition Algorithms

3.1 Principal Component Analysis (PCA)

3.1.1 How it works

The task of facial recognition is to discriminate input signals (image data) into several classes (persons). The input signals are noisy (e.g. the noise is caused by differing lighting conditions, pose etc.), yet the input images are not completely random and in spite of their differences there are patterns that occur in any input signal. Such patterns, which can be observed in all signals, could be - in the domain of facial recognition - the presence of some objects (eyes, nose, mouth) in any face as well as relative distances between these objects. These characteristic features are called eigenfaces in the facial recognition domain (or principal components generally). They can be extracted out of the original image data by means of a mathematical tool called Principal Component Analysis (PCA) [23,33].

By means of PCA each original image of the training set can be transformed into a corresponding eigenface. An important feature of PCA is that one can reconstruct any original image from the training set by combining the eigenfaces. Eigenfaces are the characteristic features of the face; the original face image can be reconstructed from eigenfaces if all the eigenfaces (features) are added up in the right proportions. Each eigenface represents certain features of the face, which may or may not be present in the original image. If the feature is present in the original image to a higher degree, then the share or "sum" of the corresponding eigenface should be greater. If the particular feature is not (or almost not) present in the original image, then the corresponding eigenface should contribute a smaller (or not at all) part to the sum of eigenfaces. So, in order to reconstruct the original image from the eigenfaces, one has to build a kind of weighted sum of all eigenfaces. That is, the reconstructed original image is equal to a sum of all eigenfaces, with each eigenface having a certain weight. This weight specifies, to what degree the specific feature (eigenface) is present in the original image.

Combining all the eigenfaces extracted from original images reconstructs the original images from the eigenfaces exactly, although it is only possible to use part of the eigenfaces. The reconstructed image is an approximation of the original image. Losses due to omitting some of the eigenfaces can be minimized by choosing the most important features (eigenfaces). The omission of eigenfaces is necessary due to scarcity of computational resources.

How does this relate to facial recognition? It is possible to both extract the face from eigenfaces given a set of weights, but also to go the opposite way. That is, it's possible to extract the weights from eigenfaces and the face to be recognized. These weights tell the amount by which the face in question differs from "typical" faces represented by the eigenfaces. Weights can determine two important things:

1. Determine if the image in question is a face at all. If the weights of the image differ by a large degree from the weights of face images (i.e. images, from which we know for sure that they are faces), the image probably is not a face.

2. Similar faces (images) possess similar features (eigenfaces) to similar degrees (weights). If one extracts weights from all the images available, the images could be grouped into clusters. That is, all images having similar weights are likely to be similar faces.

3.1.2 Face is viewed as a vector

A face is an image. This image can be considered as a vector. Suppose that the image's width is **w** and image's height is **h**. **w** is the number of pixels horizontally and **h** is the number of pixels vertically. Thus the number of pixels for each vector will be **w*h**. To construct the vector, we put all the rows of the image beside each other as shown in Fig. 3.1 and Fig. 3.2 [54].

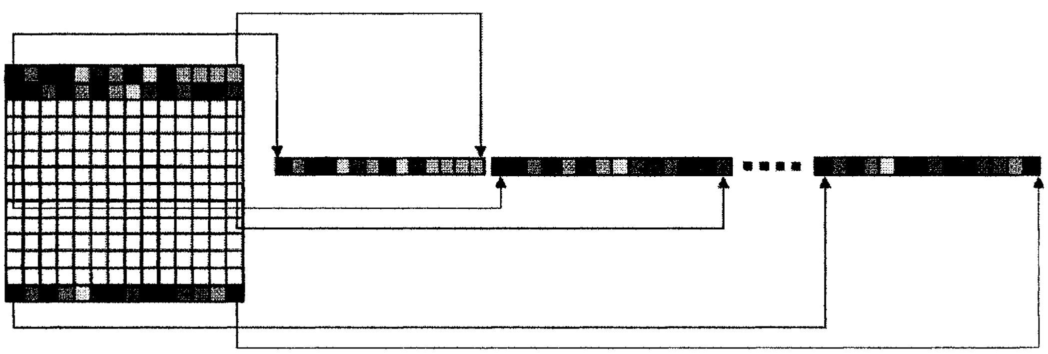


Fig. 3.1 – Construction of face vector

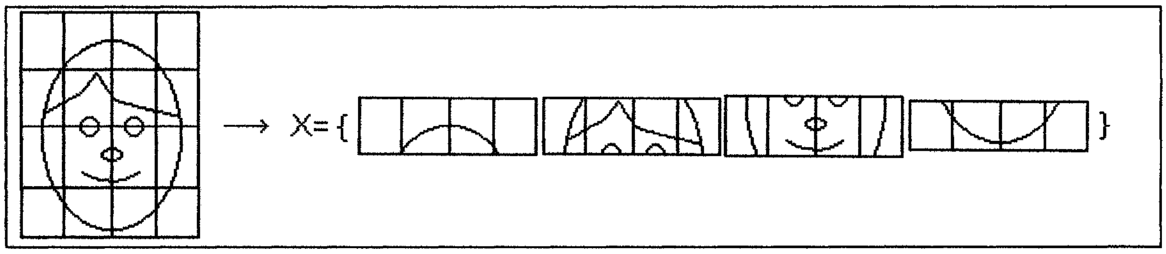


Fig. 3.2 - formation of the face's vector from the face's image

By putting rows of the images beside each other, you make a vector as shown in Fig. 3.3.

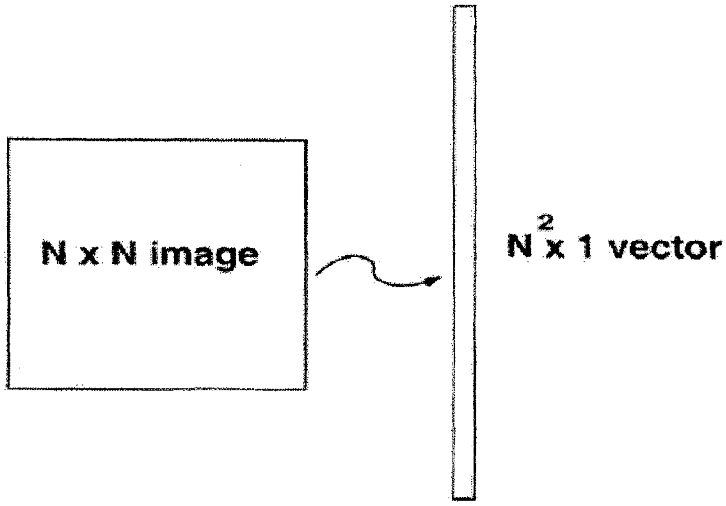


Fig. 3.3 – mapping NxN image into N^2 vector

The face vector belongs to a face space. This space is the image space, the space of all images whose dimension is **w** by **h** pixels. The basis of the image space is composed of the following vectors:

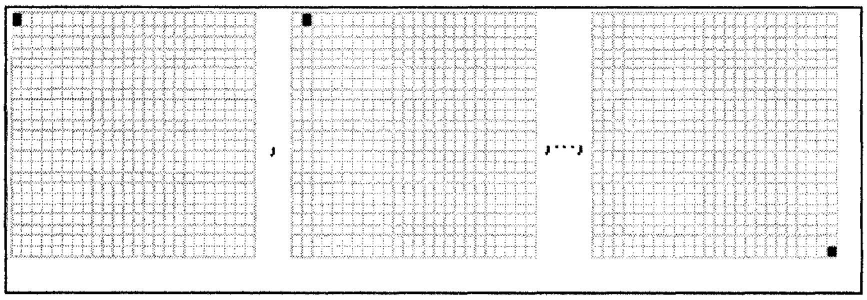


Fig. 3.4 – Image space

All the faces look like each other. They all have two eyes, a mouth, a nose, etc. located at the same place. Therefore, all the face vectors are located in a very narrow cluster in the image space, as shown in the Fig. 3.5.

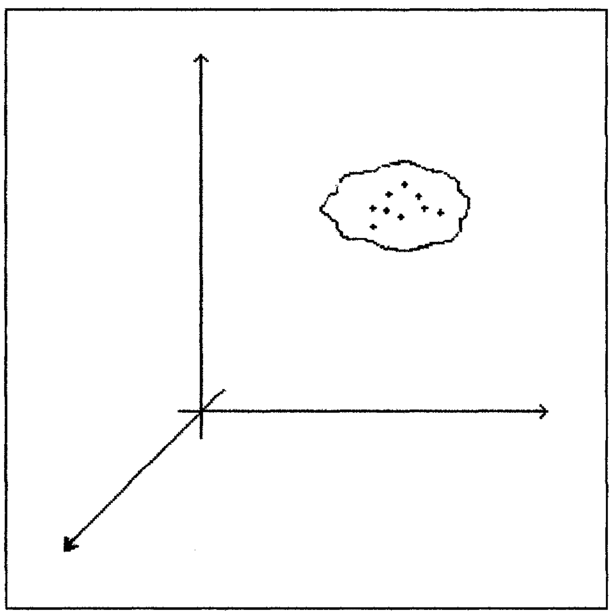


Fig. 3.5 - Image Space and face cluster

The full image space is not an optimal space for face description. The task presented here aims to build a face space that describes the faces better. The basis vectors of this face space are called the principal components (eigenfaces).

In the field of face recognition most of the common methods employed are Principal Component Analysis. Principal Component Analysis is based on the

Karhunen-Loeve (K-L), or Hotelling Transform, which is the optimal linear method for reducing redundancy, in the least mean squared reconstruction error sense. PCA became popular in face recognition with the success of eigenfaces. The idea of principal component analysis is based on the identification of linear transformation of the co-ordinates in a system: "The three axes of the new co-ordinate system coincide with the directions of the three largest spreads of the point distributions."[15]

Principal component analysis uses singular value decomposition to compute the principal components. A matrix whose rows consist of the eigenvectors of the input covariance matrix multiplies the input vectors. This produces transformed input vectors whose components are uncorrected and ordered according to the magnitude of their variance. Those components, which contribute only a small amount to the total variance in the data set, are eliminated. It is assumed that the input data set has already been normalized so that it has a zero mean.

The most important components of each face are located in a very narrow cluster. Thus the full image is not an optimal space for face recognition and there are many redundant components that are not important for face recognition. The goal of Principal Component Analysis is to reduce the dimension of the set or space. That means it aims to catch the total variation in the set of the training faces, and to explain this variation by a few variables. Understanding a few variables is always easier than understanding a huge number of variables. This is important if there is huge number of faces to be processed [8]. There are two approaches in PCA: 1) statistical approach and; 2) neural network approach. This thesis will focus on PCA using statistical approach.

3.1.3 Statistical PCA

In statistics, principal components analysis (PCA) is a technique that can be used to simplify a dataset; more formally it is a linear transformation that chooses a new coordinate system for the data set so that the greatest variance by any projection of the data set comes to lie on the first axis (then called the first principal component), the second greatest variance on the second axis, and so on. PCA can be used for reducing dimensionality in a dataset while retaining those characteristics of the dataset that contribute most to its variance by eliminating the later principal components (by a more or less heuristic decision). These characteristics may be the "most important", but this is not necessarily the case, depending on the application [40][19]. Image space is redundant so the goal of PCA is to reduce the space dimension to get fewer variables for recognition.

PCA is also called the Karhunen-Loève transform (named after Kari Karhunen and Michel Loève) or the Hotelling transform (in honor of Harold Hotelling). PCA is an optimal method that uses linear transformation for keeping the subspace with large variance. However this comes at the price of greater computational requirement, e.g. if compared to the discrete cosine transform. PCA does not have a fixed set of basis vectors and this is the difference between this method and other linear transforms. The basis vectors in PCA depend on the dataset that is used.

If we consider that the mean for the sample is zero (the sample mean of the distribution has been subtracted away from the data set), the principal component w_1 of a dataset x can be defined as:

$$w_1 = \arg\max_{\|w\|=1} E\left\{\left(w^T x\right)^2\right\}$$

The k-th component can be found by subtracting the first k-1 principal components from x:

$$\hat{x}_{k-1} = x - \sum_{i=1}^{k-1} w_i w_i^T x$$

And we can substitute this as the new dataset to find a principal component:

$$w_k = \arg\max_{\|w\|=1} E\left\{\left(w^T \hat{x}_{k-1}\right)^2\right\}$$

There is a simpler way to calculate w_i. This way uses the empirical covariance matrix of x, which is the measurement vector. When eigenvalues and eigenvectors of the covariance matrix are calculated, then we can find the eigenvectors with the largest eigenvalues correspond to the dimensions that have the strongest correlation in the dataset. Then what we found as original measurements are projected onto a reduced vector space. We have to mention that eigenvectors X are actually the columns of V, where X=ULV' is the singular value decomposition of X [56].

There are two phases in PCA: 1) The training phase and; 2) the recognition phase. In the training phase, training images are selected and PCA variables are calculated. In the recognition phase, the calculated variables are used to recognize an unidentified image. Fig. 3.6 illustrates the steps that are required in PCA algorithm [8]:

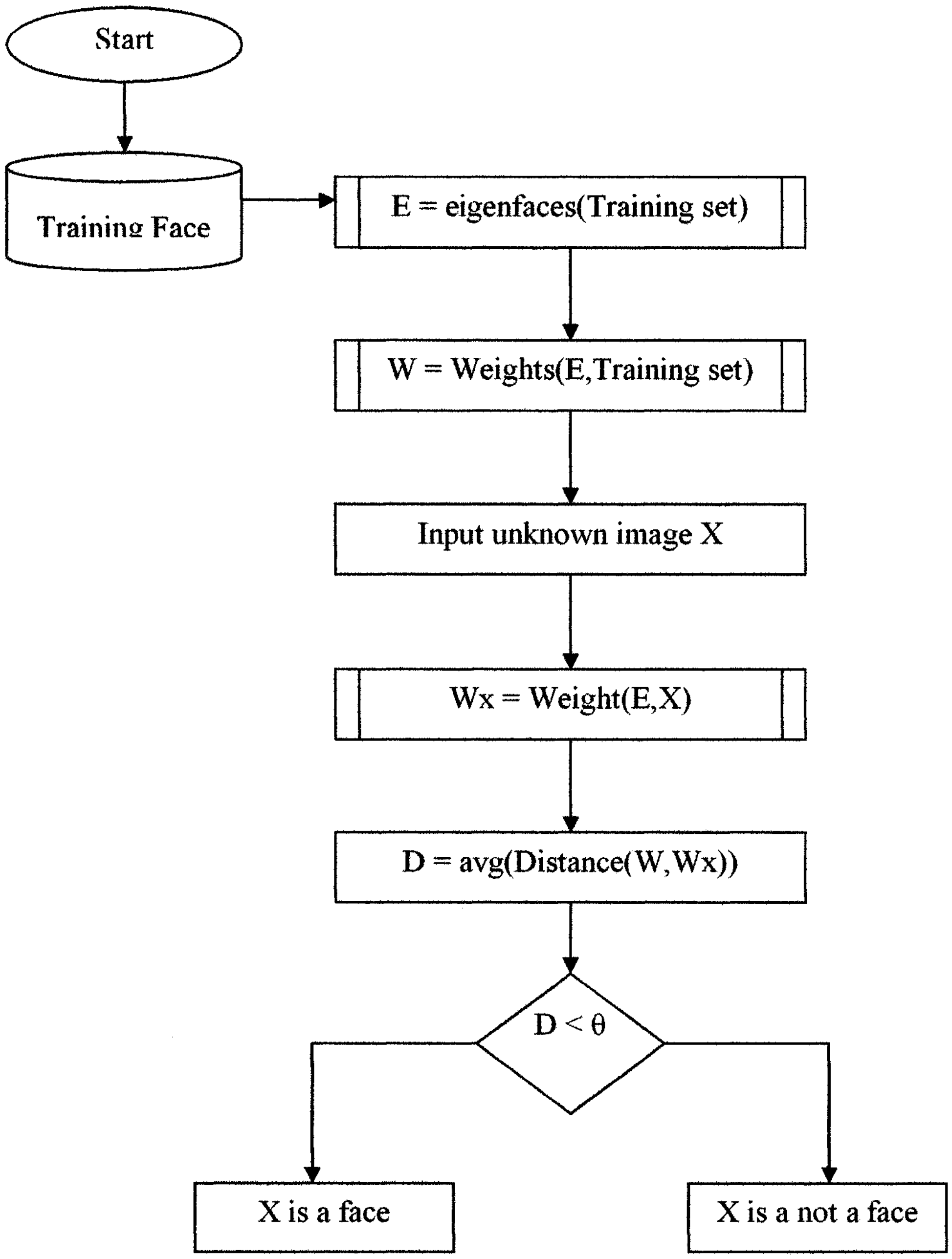


Fig. 3.6 – PCA algorithm

In this algorithm, the training set is transformed into eigenfaces. Then required variables are calculated for each image in the training set (weights). After weights are calculated, weights for an unknown image are calculated as well. In the last step, the difference between the weights in the training set and the weights for the unknown image are compared and the closest difference based on the threshold θ is considered as the recognized image.

Suppose that training image consists N pixels (N = w*h). Let $\{\Gamma_i \mid i = 1,..M\}$ be the training set of images (*M* is the number of face images in the training set). The average face of these M images is calculated by the following:

$$\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i \quad (1)$$

Then each face Γi differs from the average face Ψ by Φi

$$\Gamma_i - \Psi; i = 1,...,M \quad (2)$$

A covariance matrix of the training images can be constructed as follows:

$$C = AA^T \quad (3)$$

where $A = [\Phi_1,...,\Phi_M]$. The basis vectors of the face space, i.e., the eigenfaces, are then the orthogonal eigenvectors of the covariance matrix C.

A is $N \times M$, A^T is $M \times N$. Thus C will be an $N \times N$ matrix. Finding eigen-vectors of an $N \times N$ matrix will be a difficult task due to its size. Therefore a simplified calculation has to be adopted [15].

Since the number of training images is usually less than the number of pixels in an image, there will be only *M*-1, instead of *N*, meaningful eigenvectors. The eigenfaces are computed by first finding the eigenvectors, $v_l(l = 1,...,M)$, of the *M* by *M* matrix *L*:

$$L = A^T A \quad (4)$$

The eigenvectors, $u_l(l = 1,...,M)$, of the matrix C are then expressed by a linear combination of the difference face images, $\Phi_i(i = 1,..M)$, weighted by $v_l(l = 1,...,M)$:

$$U = [u_1,...,u_M] = [\Phi_1,...\Phi_M][v_1,...,v_M] = AV \quad (5)$$

where $[v_1,...,v_M] = \text{diag}(v_1,...,v_M)$

The first eigenface is the average face, while the rest of the eigenfaces represent variations from this average face. The first eigenface is a good face filter: each face multiplied pixel by pixel (inner product) with this average face yields a number close to one — with non–face images the inner product is far less than one. The direction of the largest variation (from the average) of the training vectors is described by the second eigenface. The direction of the second largest variation (from the average) of the training vectors is described by the third eigenface, and so on.

Each eigenface can be viewed as a feature. When a particular face is projected onto the face space, its vector (made up of its weight values with respect to each eigenface) into the face space describes the importance of each of those features in the face. Figure 3.4 describes this process pictorially.

Fig. 3.7 illustrates the conversion of training image into eignefaces, which is the first step in PCA:

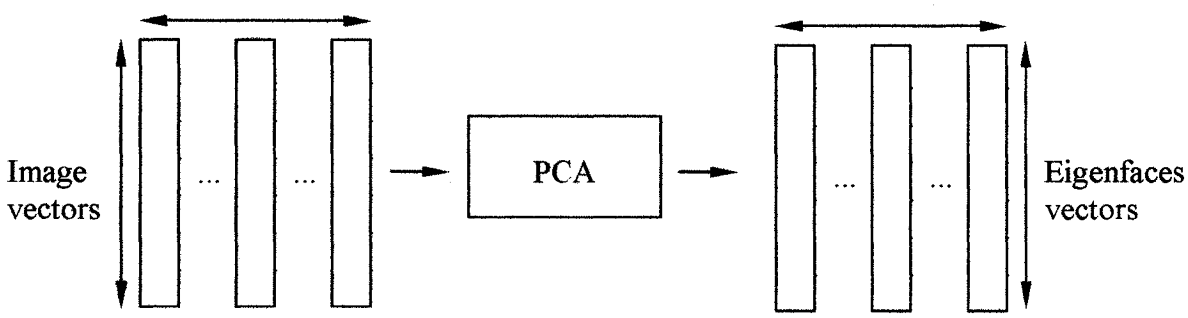


Fig. 3.7 - Eigenfaces generation process

In practice, a smaller set of M'(M'<M) eigenfaces is sufficient for face identification. Hence, only M' significant eigenvectors of L, corresponding to the largest M' eigenvalues, are selected for the eigenface computation, resulting in a further data compression. M' is determined by a threshold, θ_λ, of the ratio of the eigenvalue summation:

$$M = \min_r \left\{ r \mid \frac{\sum_{l=1}^{r} \lambda_l}{\sum_{l=1}^{M} \lambda_l} > \theta_\lambda \right\} \quad (6)$$

In the training stage, the face of each known individual, Γ_k, is projected into the face space and an M'-dimensional vector, Ω_k, is obtained:

Here we use $U' = [\Phi'_1, \ldots \Phi'_M]$

$$\Omega_k = U'^T (\Gamma_k - \Psi); \quad k = 1, \ldots, M \; (7)$$

where N_c is the number of face classes.

For comparison, two methods are used to describe a face class in the face space. The first method, referred to as the averaging representation, calculates the class vector by averaging the projected vectors from the training images of the corresponding individual [1]. The second method, the point-set representation, describes a face class by a set of vectors projected from all the training images of an individual [2].

A distance threshold, θ_c, that defines the maximum allowable distance from a face class as well as from the face space, is set up by computing half the largest distance between any two face classes:

$$\theta = \frac{1}{2} \max_{j,k} \left\{ \| \Omega_j - \Omega_k \| \right\} j, k = 1, \ldots, M \quad (8)$$

Once the eigenfaces have been computed, the face space has to be populated with known faces. Usually these faces are taken from the training set.

Each known face is transformed into the face space and its components stored in memory.

At this stage the identification process can begin. An unknown face is presented to the system. The system projects it onto the face space and computes its distance from all the stored faces. The face is identified as being the same individual as the face which is nearest to it in face space. There are several methods of computing the distance between multidimensional vectors. Here, a form of Euclidean distance is chosen. In fact it is the cosine of the angle between the two faces which is computed. Since the faces are normalized (vector magnitude equals 1), a comparison of cosine distances and Euclidean distances is identical. The advantage is that cosine distances can be calculated more efficiently.

In the recognition stage, a new image, Γ, is projected into the face space to obtain a vector, Ω:

$$\Omega = U^T \Gamma - \psi \quad (9)$$

The distance of Ω to each face class is defined by

$$\varepsilon_k^2 = \| \Omega - \Omega_k \|^2; k = 1,\ldots,M \quad (10)$$

For the purpose of discriminating between face images and non-face like images, the distance, ε, between the original image, Γ_f, and its reconstructed image from the eigenface space, Γ_f, is also computed:

$$\varepsilon^2 = \| \Gamma - \Gamma_f \|^2 \quad (11)$$

where

$$\Gamma_f = U.\Omega + \Psi \quad (12)$$

These distances are compared with the threshold given in equation (8) and the input image is classified by the following rules:

- IF $\varepsilon \geq \theta_c$

  THEN input image is not a face image;
- IF $\varepsilon < \theta_c$ AND $\forall k, \varepsilon_k \geq \theta_c$

  THEN input image contains an unknown face;
- IF $\varepsilon < \theta_c$ AND $\varepsilon_{k*} = \min_k \{\varepsilon_k\} < \theta_c$

  THEN input image contains the face of individual $k*$

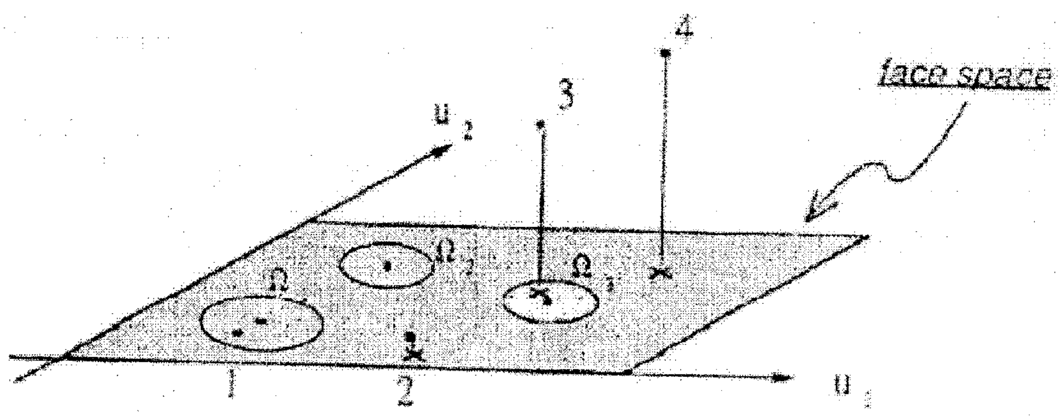


Fig. 3.8 – Face space[1]

[1] Taken from [27]

The following figure shows eigenfaces for a sample dataset:

Fig. 3.9 – Eigenfaces for a sample dataset[2]

3.2 *Fisher's Linear Discriminant (FLD)*

The Fisherface [35] method was suggested by Belhumeur et al. [35]. Both the Eigenface algorithm and the Fisherface methods project images into a feature space. However, Fisherface uses Fisher's Linear Discrimination (FLD), a class-specific method.

[2] Taken from [22]

FLD, on the other hand, tries to find a projection, which separates data clusters:

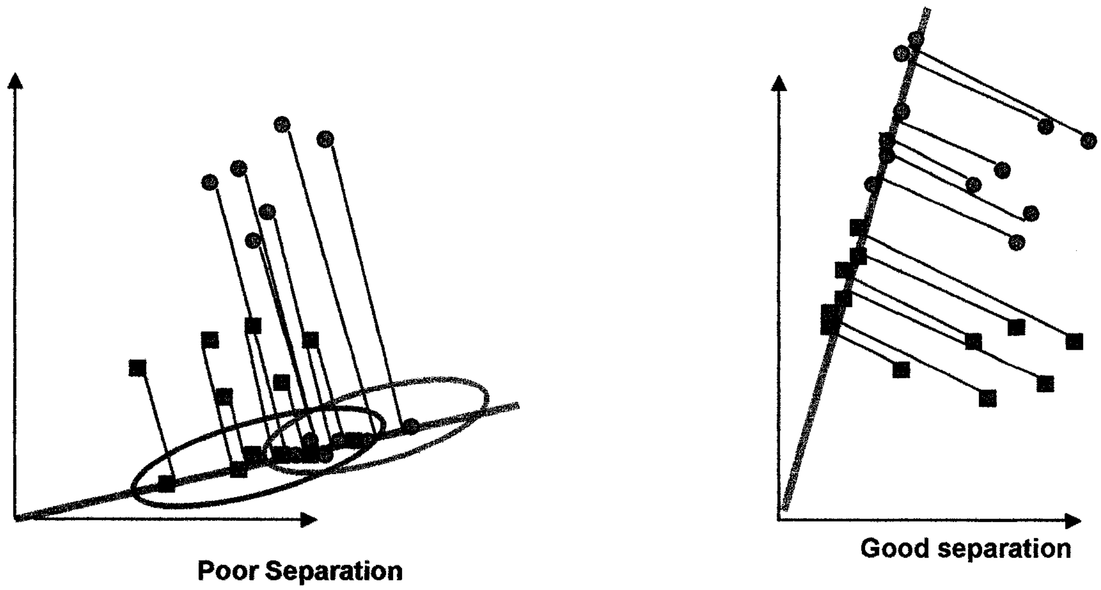


Fig. 3.10 – Cluster Separation

In PCA, projection is best for reconstruction of images from a low dimensional basis. However, the projection does not make use of the between-class variance. The projection may not be optimal for discrimination for different classes. In FLD, the projection maximizes the ratio of the between-class scatter to that of the within-class scatter. It tries to reshape the scatter to make it more reliable for classificatioin [35]. The between-class scatter is defined as following:

$$S_B = \sum_{k=1}^{c} m_i (\Psi_k - \Psi)(\Psi_k - \Psi)^T \quad (13)$$

where c is the number of classes and m_i is the number of samples in class T_i. The within-class scatter matrix is defined as:

$$S_W = \sum_{i=1}^{c} \sum_{X_k \in C_i} (X_k - \Psi_i)(X_k - \Psi_i)^T \quad (14)$$

The following figure shows the attempt to make a separation by calculating S_B and S_W:

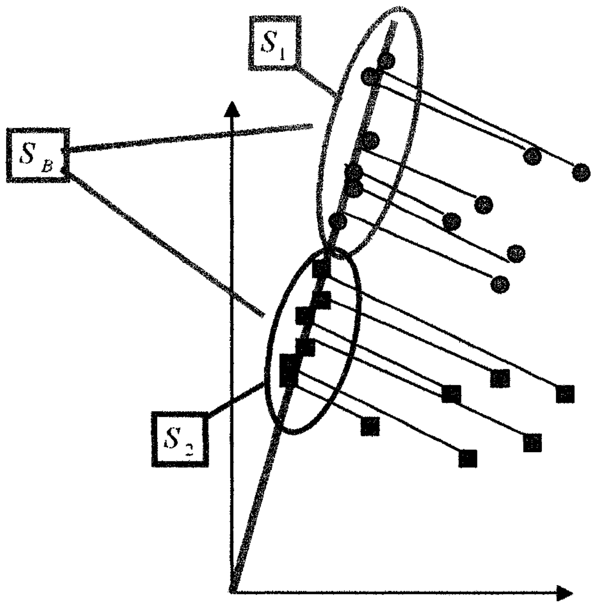


Fig. 3.11 – Scatter matrix S_B and S_W

where Ψ_i is the mean image of class C_i. The optimal projection W_{opt} is chosen as the matrix with orthonormal columns, which 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. It is defined as follows:

$$W_{opt} = \arg\max_{W} \frac{|W^T S_B W|}{|W^T S_W W|} \quad (15)$$

This equation is trying to find the generalized eigenvectors (V) and eigenvalues (Λ) by solving:

$$S_B V = \Lambda S_W V \text{ (16)}$$

This is exactly what was done in (5) PCA. That means the U_{fld} matrix can be constructed as following:

$$U_{fld} = [u_1,...,u_M] = [\Phi_1,...\Phi_M][v_1,...,v_M] = A.V \text{ (17)}$$

The difference in the PCA procedure is that you do not subtract from each image the average image. The average projected coefficients of each person Ω_i and U_{fld} are stored.

$$\Omega_i = U_{fld}^T \varphi_i \text{ (18)}$$

To identify a face, φ, calculate:

$$\theta = U_{fld}^T \varphi \text{ (19)}$$

Then the shortest distance between θ and average projected coefficient Ω_i is calculated. This will be the recognized image:

$$s = \arg_{1 \le i \le m} \min \{\| \theta - \Omega_i \|\} \text{ (20)}$$

Advantages and disadvantages of FLD

- Reduce dimension of the data from *N2* to *P-1*
- Can outperform eigenfaces on a representative Database

- Works also with various illuminations etc.
- Can only classify a face which is "known" to DB

3.3 Differences between PCA and FLD

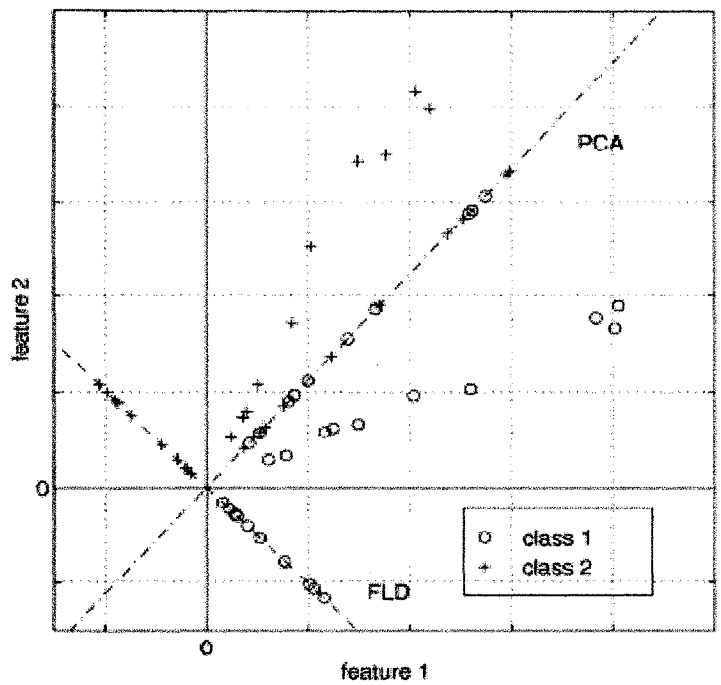


Fig. 3.12 - A comparison of PCA and FLD when c=2

The above figure briefly shows the difference between the eigenface method and the fisherface method [27]. It is easy to notice that PCA smears the classes together so that they are no longer separable, but it achieves larger scatter, which means easier classification. The Fisher Linear Discriminant (FLD) method is different from the eigenface (PCA) method in a sense that W_{opt} is chosen as the matrix with orthonormal columns, which 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.

PCA is a popular technique in pattern recognition. However, PCA is not optimized for class separability. An alternative is the linear discriminant analysis, which does take this separability into account. PCA optimally minimizes reconstruction error under the L2 norm. The problem with the eigenface (PCA) method is that the scatter is maximized not only in between-class but also within class as shown in the following figure:

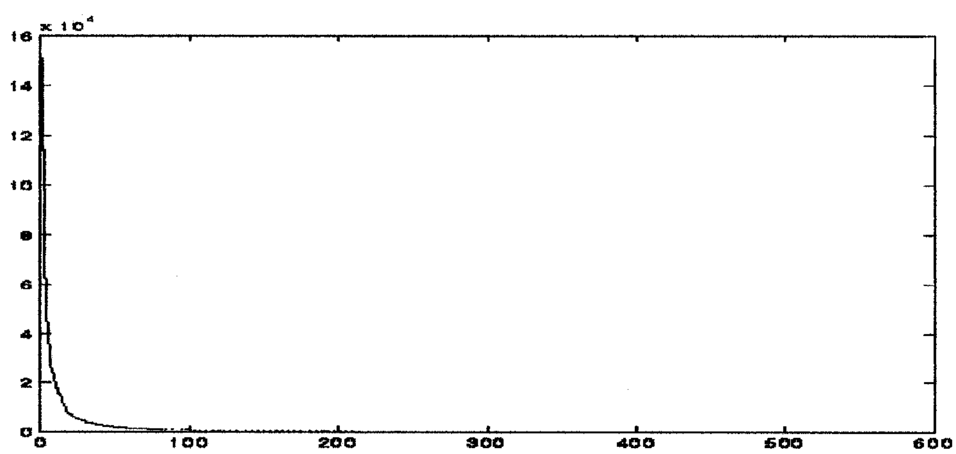


Fig. 3.13 - A graph showing eigenvalue curve drops very quickly

In general FLD performs better than PCA. The followings are the advantages of FLD over PCA in terms of performance:

- FLD is faster than PCA because of better projection
- Has lower error rates
- Works well even in different illumination

- Works well even with different facial expressions.

3.4 *Small sample size problem*

In face recognition tasks, the dimension of the sample space is typically larger than the number of the samples in the training set. So the within-class scatter matrix is singular. This problem is known as the "small sample size" problem. To overcome this problem, a new method has been proposed called Discriminative Common Vector method, which is based on a variation of Fisher's Linear Discriminant Analysis for small sample size [13]. This algorithm uses within-class scatter matrix to produce common vectors. Then the common vectors are used for classification of new faces. This method claims more accuracy, efficiency and stability comparing to traditional methods like PCA and FLD.

3.4.1 Problems with PCA

PCA is unsupervised since it does not consider the classes within the training set data. In choosing a criterion that maximizes the total scatter, this approach tends to model unwanted within-class variations such as those resulting from the differences in lighting, facial expression, and other factors [49, 50]. Also, because the criterion does not minimize the within-class variation, there could be overlap in the result compared to other methods. Thus, the projection vectors chosen for optimal reconstruction may obscure the existence of the separate classes.

3.4.2 Problems with FLD

This method (FLD) solves the limitation of the Eigenface method by applying Fischer's Linear Discriminant criterion as mentioned below:

$$J_{FLD}(W_{opt}) = \arg_W \max \frac{|W^T S_B W|}{|W^T S_W W|}$$

where S_B is the between-class scatter matrix, and S_W is the within-class scatter matrix. By applying this method, the projection directions maximize the Euclidean distance between the face images of different classes on the one hand and on the other minimize the distance between the face images of the same class. The problem with this method is that it cannot be applied since the dimension of the sample space is typically larger than the number of samples in the training set. Many algorithms were proposed to fix this problem with Discriminative Common Vector approach being one of them [46].

3.4.3 Discriminative Common Vector Approach

This approach is used when the number of samples in each class is less than or equal to the dimensionality of the sample space. In this approach, the common properties of classes in the training set are extracted by eliminating the differences of the samples in each class. It is done by removing all the features that are in the direction of the eigenvectors corresponding to the nonzero eigenvalues of the scatter matrix of its own class. These common vectors are then used for recognition. In this case, instead of using a given class's own scatter matrix, a within-class scatter matrix of all classes was used to obtain the common vectors [46].

3.4.3.1 Training Stage

In order to obtain Discriminative Common Vectors, we use the range space of S_W. Suppose that the training set consists of C classes and each class has N samples. Also define x_m^i be a d-dimensional column vector which donates the mth sample from the ith class. There will be a total of M = NC samples in the training set and S_W, S_B and S_T are defined as:

$$S_W = \sum_{i=1}^{c} \sum_{m=1} \left(x_m^i - \mu_i\right)\left(x_m^i - \mu_i\right)^T \quad (21)$$

$$S_B = \sum_{i=1}^{C} N(\mu_i - \mu)(\mu_i - \mu)^T \quad (22)$$

$$S_T = \sum_{i=1}^{C} \sum_{m=1}^{N} (x_m^i - \mu)(x_m^i - \mu)^T = S_W + S_B \quad (23)$$

where μ is the mean of all samples, and μ_i is the mean of samples in the ith class. We call S_W the within-class scatter matrix and S_B the between-class scatter matrix.

Let λ_k and v_k be the kth nonzero eigenvalue and the corresponding eigenvector of $A^T A$, where $k \le M - C$. Then $\alpha_k = A v_k$ will be the eigenvector that corresponds to the kth nonzero eigenvalue of S_W. Then we obtain the same unique vector for all samples of the same class,

$$x_{com}^{i} = x_{m}^{i} - QQ^{T}x_{m}^{i} = \overline{QQ}^{T}x_{m}^{i} \quad (24)$$

$$m = 1,\ldots,N$$

$$i = 1,\ldots,C$$

where Q is the QR decomposition of S_T

Now compute the eigenvectors w_k of S_{com} corresponding to the nonzero eigenvalues, by using the matrix $A_{com}^{T}A_{com}$, where $S_{com} = A_{com}A_{com}^{T}$. There are at most C-1 eigenvectors that correspond to the nonzero eigenvalues. Use these eigenvectors to form the projection matrix $W = [w_1 \ldots w_{C-1}]$, which will be used to obtain feature vectors. These feature vectors are used in the recognition stage.

3.4.3.2 Recognition Stage

Since the optimal projection vectors w_k belong to the null space of S_W, it follows that when the image samples x_m^i of the ith class are projected onto the linear span of the projection vectors w_k, the feature vector $\Omega_i = [<x_m^i, w_1> \ldots <x_m^i, w_{C-1}>]^T$ of the projection coefficients $<x_m^i, w_k>$ will also be independent of the sample index m. Thus,

$$\Omega_i = W^T x_m^i, \quad m = 1,\ldots,N, \quad i = 1,\ldots,C$$

The feature can be called Ω_i Discriminative Common Vectors and they will be used for classification of face images.

To recognize a test image x_{test}, the feature vector of this test image is found by

$$\Omega_{test} = W^T x_{test}$$

This test feature will be compared to the common vector Ω_i of each class by calculating Euclidean distance. The closest discriminative common vector Ω_{test} is the identified image [46].

3.5 Laplacianfaces

This algorithm is an appearance-based face recognition method. By using Locality Preserving Projections (LPP), the face images are mapped into a face subspace for analysis. PCA and FLD effectively see only the Euclidean structure of face space and do not consider essential face manifold structure. But LPP finds an embedding that preserves local information, and obtains a face subspace that best detects the essential face manifold structure. In this method, unwanted variations resulting from changes in lighting, facial expression, and pose may be eliminated or reduced [52, 43].

3.5.1 Learning a Locality Preserving Subspace

As said before, PCA and FLD aim to preserve the global structure but in many real-world applications, the location structure is more important. An algorithm is proposed to achieve this, which is called Locality Preserving

Projection or LPP for learning a locality preserving subspace. The objective of this algorithm is to minimize the followings:

$$\min \sum_{ij} (y_i - y_j)^2 S_{ij} \quad (25)$$

where y_i is the one-dimensional representation of x_i and S is a similarity matrix. A possible way of defining S is as follows:

$$S_{ij} = \begin{cases} \exp(-\| x_i - x_j \|^2 / t), \| x_i - x_j \|^2 < \varepsilon \\ 0 \qquad otherwise \end{cases} \quad (26)$$

where ε is sufficiently small and greater than zero. ε defines the radius of the local neighborhood. In other words, it defines the "locality". The symmetric weights S_{ij} incurs a heavy penalty if neighboring points x_i and x_j are mapped far apart, i.e., if $(y_i - y_j)^2$ is large. Minimizing it is an attempt to ensure that, if x_i and x_j are close, then y_i and y_j are close as well.

Suppose that $X = [x_1, x_2, ..., x_n]$, and D is a diagonal matrix; its entries are column sums of S, $D_{ii} = \sum_j S_{ji}$. $L = D - S$ is the Laplacian matrix. Matrix D provides a natural measure on the data points. The bigger the value D_{ii} is, the more "important" is y_i. Thus the following constraint should be applied:

$$\begin{aligned} y^T D y = 1 \\ w^T X D X^T w = 1 \end{aligned} \quad (27)$$

The transformation vector w that minimizes the objective function is given by the minimum eigenvalue solution to the generalized eigenvalue problem:

$$XLX^T w = \lambda XDX^T w \quad (28)$$

Two matrices XLX^T and XDX^T are both symmetric and positive semi definite since the Laplacian matrix L and the diagonal matrix D are both symmetric and positive semi definite.

The Laplacian matrix for finite graph is analogous to the Laplace Beltrami operator on compact Riemannian manifolds. While the Laplace Beltrami operator for a manifold is generated by Riemannian metric, for a graph it comes from the adjacency relation. [52] showed that the optimal map preserving locality can be found by solving the following optimization problem on the manifold:

$$\min_{\|f\|_{L_2(M)}=1} \int_M \| \nabla f \|^2 \quad (29)$$

which is equivalent to

$$\min_{\|f\|_{L_2(M)}=1} \int_M L(f) f \quad (30)$$

where L is the Laplace Beltrami operator on the manifold. Thus, the optimal f has to be an eigenfunction of L. If f is linear, we have $f(x) = w^T x$. Using spectral graph theory, the integral can be approximated by $w^T XLX^T w$ and the L_2 norm of f can be approximated by $w^T XDX^T w$, which will ultimately lead to the following eigenvalue problem:

$$XLX^T w = \lambda XDX^T w \quad (31)$$

Matrix w is called eigenmaps and will be used for recognition stage.

4. Gabor Wavelet

A Gabor wavelet is a complex planar wave restricted by a two-dimensional Gaussian envelope. Gabor filters allow local frequency information to be extracted from an image. They estimate the strength of certain frequency bands and orientations at each location in the image, giving a result in the spatial domain. Gabor filters are a popular tool for image analysis, and have found widespread use in computer vision. However, Gabor filtering involves a high computational load, as the application of a Gabor filter to an image involves the convolution of the image with a set of Gabor wavelets consisting of wavelet kernels of different wavelengths and orientations.

Gabor Wavelets are used in image analysis because of their biological relevance and computational properties. They illustrate the characteristics of spatial locality and orientation selectivity and are optimally localized in the space and frequency domains [6]. The following equation defines Gabor wavelet:

$$\Psi_{\mu,\nu}(z) = \frac{\| k_{\mu,\nu} \|}{\sigma^2} e^{-\frac{\|k_{\mu,\nu}\|^2 \|z\|^2}{2\sigma^2}} \left[e^{ik_{\mu,\nu_z}} - e^{-\frac{\sigma^2}{2}} \right] \quad (32)$$

μ and v are orientation and scale of Gabor kernels, z=(x,y), ||.|| is norm operation, and the vector $k_{\mu,v}$ is calculated as follows:

$$k_{\mu,v} = k_v e^{i\Phi_\mu} \quad (33)$$

where $k_v = k_{\max} / f^v$ and $\Phi_\mu = \pi\mu/8$

$k_{\max}$ is the maximum frequency, and f is the spacing factor between kernels in the frequency.

Each kernel is a product of a Guassian envelope and a complex plane wave. The first part in brackets in Eq. 31 is the oscillatory part of the kernel and the second term is DC value. If the parameter σ has large values, then DC value is negligible. In most cases Gabor wavelets are used with the following sets:

$$v \in \{0,\ldots,4\}$$
$$\mu \in \{0,\ldots,7\}$$

4.1 Gabor Features

If $I(x,y)$ is the gray level distribution of an image, the convolution of image I and a Gabor kernel $\Psi_{\mu,v}$ is defined as follows:

$$O_{\mu,v}(z) = I(z) * \Psi_{\mu,v}(z) \quad (34)$$

where $z = (x,y)$, * is the convolution operation, and $O_{\mu,v}(z)$ is the result of the equation, which is the Gabor kernel at orientation μ and scale v. The result of the equation using different values for μ and v represents Gabor wavelet of image $I(z)$. To encompass different spatial frequencies (scales), spatial localities, and orientation selectivity, all the results have to be concatenated and an augmented feature vector X has to be built. Prior to that, the size of each $O_{\mu,v}(z)$ has to be reduced by using factor ρ. This also normalizes it to zero mean and unit variance. Then a vector is constructed by concatenating rows. This is similar to what was done in section 3.1.

The dimensionally reduced vector, normalized to zero mean and unit variance is defined as follows:

$$X^{(\rho)} = \left(O_{0,0}^{(\rho)} O_{0,1}^{(\rho)} \ldots O_{4,7}^{(\rho)}\right)^t \quad (35)$$

where t is the transpose operator. The augmented Gabor feature vector thus encompasses all the elements of the Gabor wavelet representation of the following set:

$$S = \{O_{\mu,v}(z) : \mu \in \{0,\ldots 7\}, v \in \{0,\ldots 4\}\} \quad (36)$$

Fig. 4.1 shows the Gabor wavelets of five different scales, and eight orientations: [19]

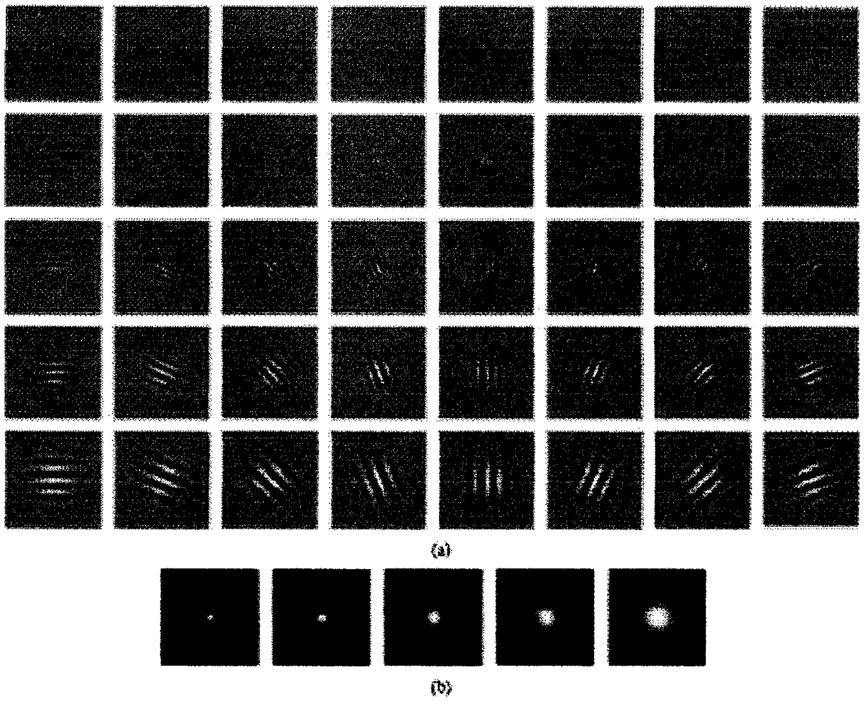


Fig. 4.1 – Gabor Wavelets. (a) The real part of the Gabor kernels at five scales and eight orientations with the following parameters: $\sigma = 2z, k_{\max} = z/2$, and $f = \sqrt{2}$. (b) The magnitude of the Gabor kernels at five different scales. The kernels exhibit desirable characteristics of spatial frequency, spatial locality, and orientation selectivity

5. Color Information in Face Recognition

One of the key challenges in face perception is to identify the role of different cues for face identification. This thesis focuses on the role of colors in face recognition. Although color appears to be a salient attribute of faces, some research has suggested that it confers little recognition advantage for identifying people. In this thesis, some experimental results are presented to show that color cues could play a significant role in face recognition. Results from this study indicate that recognition performance with color images is significantly better than that with grayscale images.

In previous studies, color was not considered an important cue in recognizing faces because in experiments people were able to accurately identify faces that were artificially colored, even in hot pink and electric blue. The clear perception of colors in the environment illustrates that color must be an important factor that allows people to interpret complex scenes and recognize objects although this is a topic of debate among researchers. Most of the theoretical modeling and experimental work in recognition is based on shape cues. A long-standing theory is that objects may be encoded in terms of their luminance-defined bounding edge structure [48][17] with surface properties such as color and texture being of little consequence. Other researchers have proposed models of recognition that rely strongly on color information [29].

A small body of research has been done on the contribution of color in face recognition. A notable study conducted by Kemp and his colleagues [42] found that observers were able to process normally even those faces that had been subjected tasks included recognizing familiar faces or spotting differences between faces. Color appeared to confer no significant recognition advantage beyond the luminance information. In explaining this data, Bruce and Young [47] suggested that the lack of a contribution of color cues to face recognition is because it does not affect shape from shading processes, which are believed to be largely 'color-blind' [36].

Another reason that some researchers discount color as a contributing factor is that in some situations strong shape cues are available (as in high resolution face images), and performance is already at the highest level making colors contribution unclear. These researchers argue that where the quality of the image is poor and shape cues are progressively degraded, the role of colors could be more obvious. Most face recognition systems work with gray-scaled images. Color could play an important role in face recognition by adding more discriminative features to the images and the recognition system. To use color features in face recognition, the following system is proposed [29].

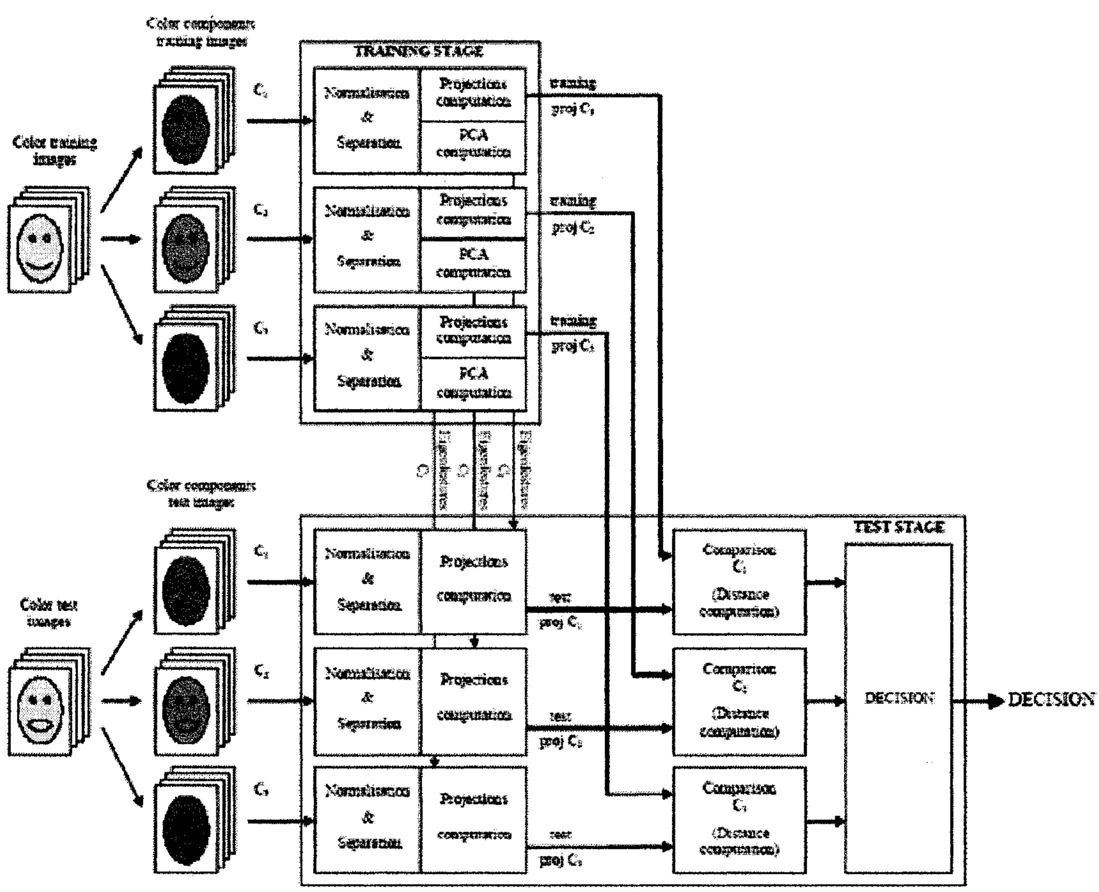


Fig. 5.1 – Proposed recognition system for Color images

The process is done in two stages: 1) Training and; 2) Recognition. The first step of each stage is to split a color image into three different images. The first image contains the information about the red color, the second image contains the information about the green color and the last image contains the information about the blue color. Each pixel in each image has the value from 0 to 255. The value represents the contrast for each color. In the next step, each separate image (in the training set) is fed into a face recognition method to calculate the required components and each image is then treated as a gray-scaled image. All the variables are calculated and stored. After calculating the variables, the training stage is complete.

In the recognition stage, the same procedure is applied to the recognition image, where each separate image is fed into the recognition stage. After the recognition is completed, three distance computations (one for Red, one for Green and one for Blue) are produced. The result of distance computation is then fed into a decision making component. The global distance between the test and the training image is the weighted sum of the different distances so that the contribution of each eigen feature to the recognition stage is the same, and uses each color component well. Weighting factors may be altered to give more importance to certain features than others. This is important for the color information as different weights can be given to the different color components as well as to the different eigenfeatures, thus emphasizing each one appropriately.

This evidence of color's contribution to face recognition brings up an interesting question regarding the specific role it plays. One possibility is that color provides diagnostic information - the precise hue of a person's hair or skin may allow us to identify them. Color may also facilitate low-level image analysis, and thus indirectly aid face recognition. An example of a low-level task is image segmentation - determining where one region ends and the other starts. As many years of work in computer vision has shown [Haralick, 1985; Felzenszwalb and Huttenlocher, 1998], this task is notoriously difficult and becomes even more intractable as images are degraded. Color may facilitate this task by supplementing the luminance-based cues and thereby lead to a better parsing of a degraded face image in terms of its constituent regions

6. Experiments Using CVL Database

A color database (**CVL Face Database**) has been selected for the experiments. It has been prepared by Peter Peer, Faculty of Computer and Information Science, University of Ljubljana. The database has the following features:

- 114 persons
- 7 images for each person
- resolution: 640*480 pixels
- format: jpeg
- images taken with Sony Digital Mavica under uniform illumination, no flash and with projection screen in the background
- persons age: mostly around 18 (pupils and some professors)
- persons sex: mostly male (around 90%)
- public domain

From this database, 15 individuals have been selected. There are 7 images per individual in different poses. For the training stage, the first 6 images for each individual have been selected. The last image was used for the recognition stage. Appendix I illustrates the selected set for training and recognition. There are 2 women and 13 men. This set is used for different experiments. Experiments have been done using the mentioned data set. To understand the effect of color in face recognition, the experiments have been done using color images and gray-scaled images. Gray-scaled images have been constructed from the color images dataset. The following formula was used to convert color images to gray-scaled images:

```
Color.FromArgb((R+G+B)/3,(R+G+B)/3,(R+G+B)/3)
```

Here R represents the value for Red color, G represents the value for Green color and B represents the value of Blue color. These values are between 0 and 255. Average for R, G and B is calculated and is passed to FromArgb. This function returns the gray-scaled color for a specific pixel. The image is processed pixel by pixel until all the pixels are converted to gray-scaled. Appendix II illustrates the converted gray-scaled dataset using the original color dataset.

Four different methods for face recognition were used for these experiments: PCA, FLD, Discriminative Common Vectors and Laplacianfaces. Gabor filters are also applied to some of them to see if recognition is improved. Here is the list of experiments performed:

1) PCA using gray-scaled dataset

2) FLD using gray-scaled dataset

3) PCA using color dataset

4) FLD using color dataset

5) PCA using color dataset + Gabor filters

6) FLD using color dataset + Gabor filters

7) Discriminative Common Vectors using gray-scaled dataset

8) Discriminative Common Vectors using color dataset

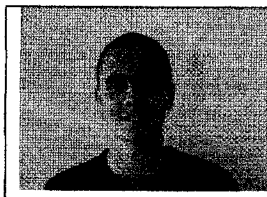
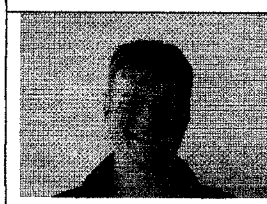
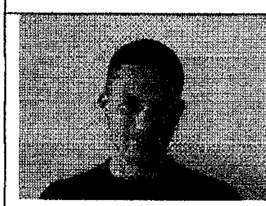
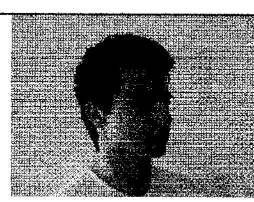
9) Laplacianfaces using gray-scaled dataset

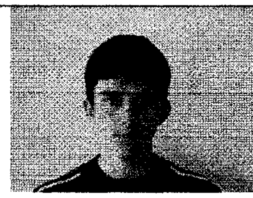
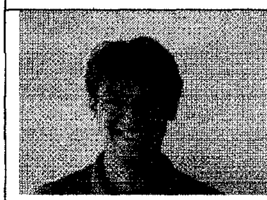
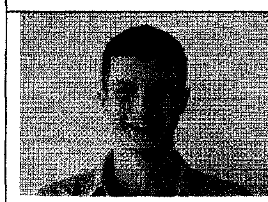
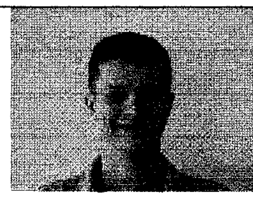
10)Laplacianfaces using color dataset

6.1 *PCA using gray-scaled dataset*

The results are shown in the following table:

Recognition Image	Recognized as	Recognized correctly?
		Yes
		Yes
		No

		Yes
		Yes
		No
		Yes
		Yes
		No
		Yes

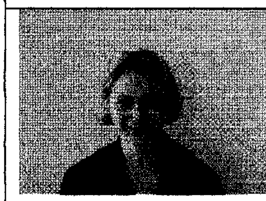
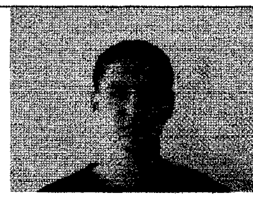
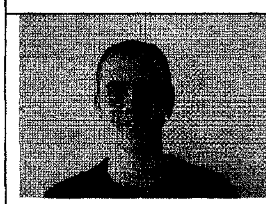
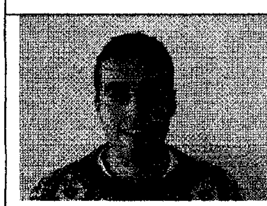
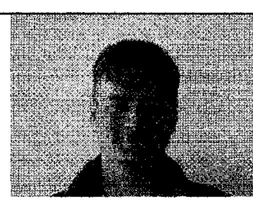
		Yes
		Yes
		No
		Yes
		Yes

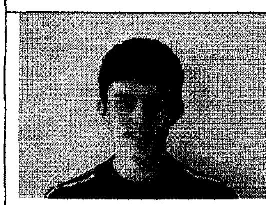
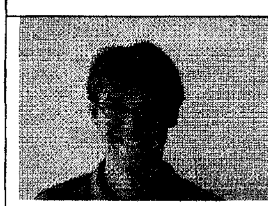
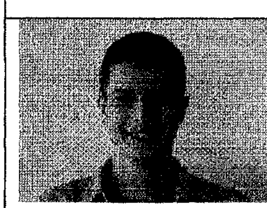
The recognition rate is 11/15 = 0.73

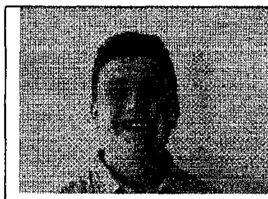
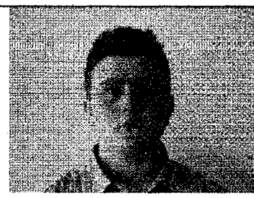
6.2 FLD using gray-scaled dataset

The results are shown in the following table:

Recognition Image	Recognized as	Recognized correctly?

		Yes
		Yes
		No
		Yes
		Yes
		Yes
		No

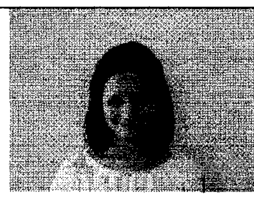
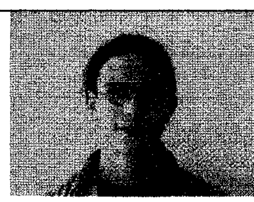
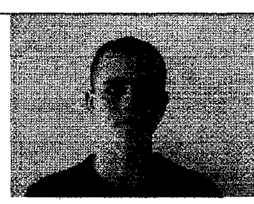
		Yes
		Yes
		Yes
		Yes
		Yes
		No
		Yes

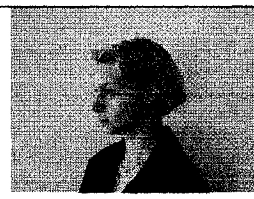
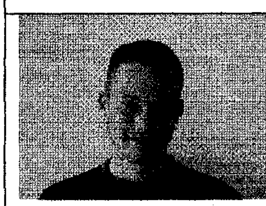
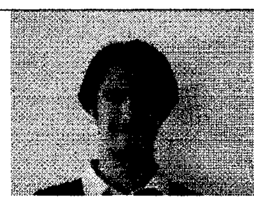
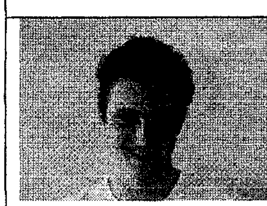
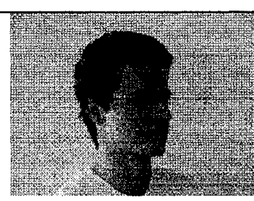
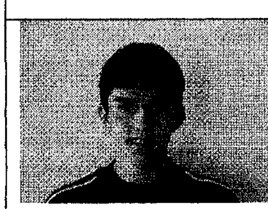
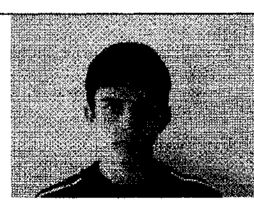
		Yes

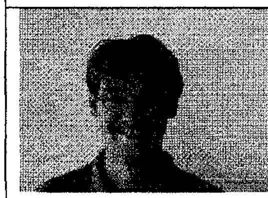
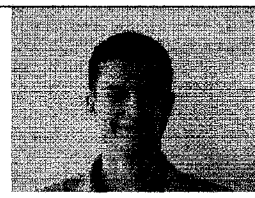
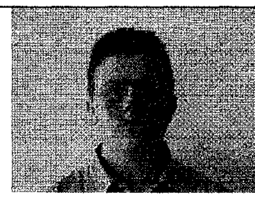
The recognition rate is 12/15 = 0.80

6.3 *PCA using gray-scaled dataset + Gabor Filters*

The results are shown in the following table:

Recognition Image	Recognized as	Recognized correctly?
		Yes
		Yes
		Yes
		Yes

		Yes
		No
		Yes
		Yes
		Yes
		Yes
		Yes

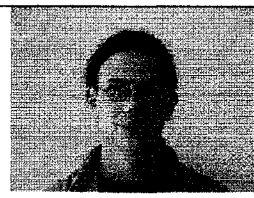
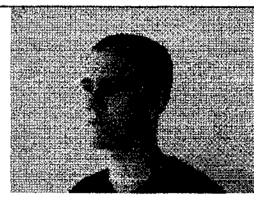
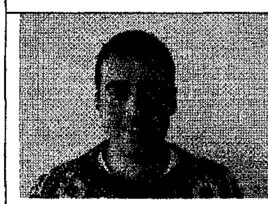
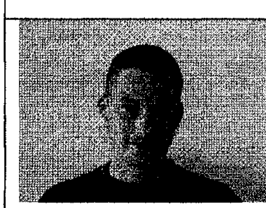
		Yes
		No
		Yes
		Yes

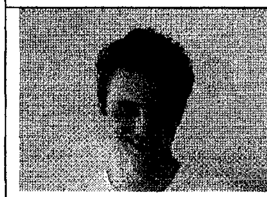
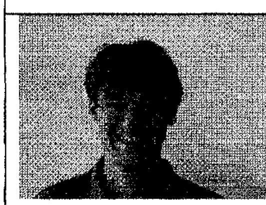
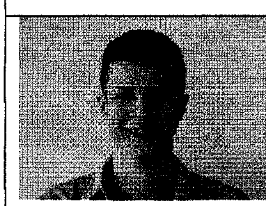
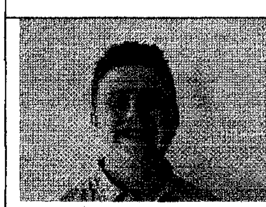
The recognition rate is 13/15 = 0.86

6.4 FLD using gray-scaled dataset + Gabor Filters

The results are shown in the following table:

Recognition Image	Recognized as	Recognized correctly?
		Yes

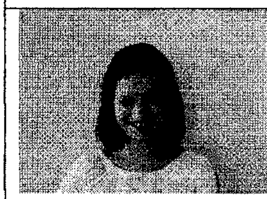
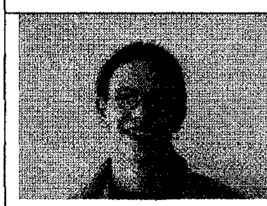
		Yes
		Yes
		Yes
		Yes
		Yes
		No
		Yes

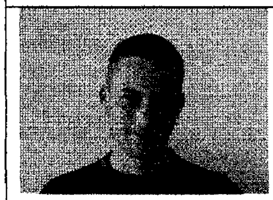
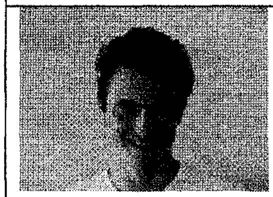
		Yes
		Yes
		Yes
		Yes
		No
		Yes
		Yes

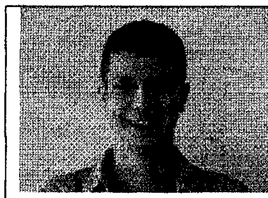
The recognition rate is 13/15 = 0.86

6.5 *Discriminative Common Vectors using gray-scaled dataset*

The results are shown in the following table:

Recognition Image	Recognized correctly?
	Yes
	Yes
	Yes
	Yes
	Yes
	No

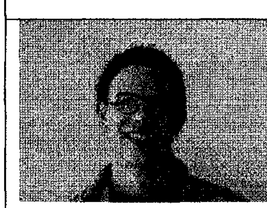
	Yes
	Yes
	Yes
	Yes
	Yes
	Yes
	Yes

	Yes
	Yes

The recognition rate is 14/15 = 0.93

6.6 *Laplacianfaces using gray-scaled dataset*

The results are shown in the following table:

Recognition Image	Recognized correctly?
	Yes
	Yes
	No

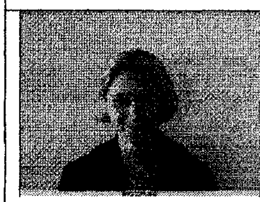
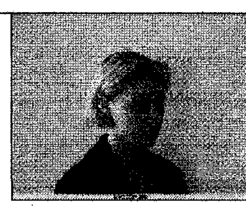
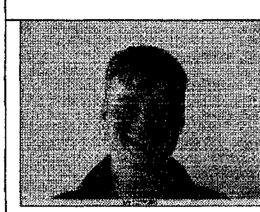
	Yes
	Yes
	Yes
	Yes
	Yes
	Yes
	Yes

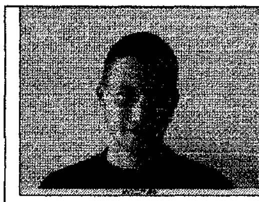
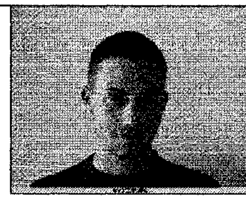
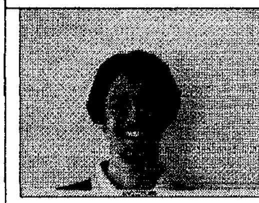
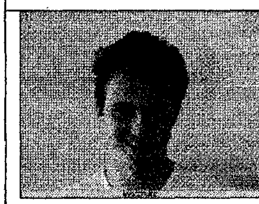
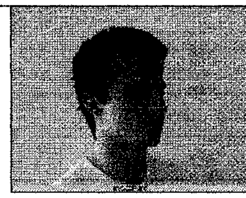
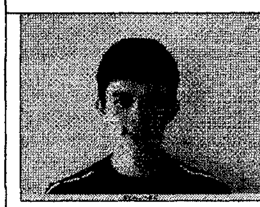
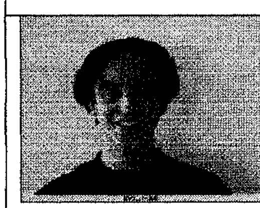
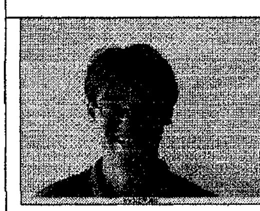
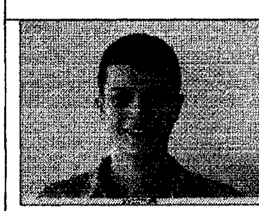
	Yes
	Yes
	Yes
	Yes
	Yes

The recognition rate is 14/15 = 0.93

6.7 *PCA using color dataset*

The results are shown in the following table:

Recognition Image	Recognized as	Recognized correctly?
		Yes
		Yes
		No
		Yes
		Yes
		No
		Yes

		Yes
		Yes
		Yes
		Yes
		Yes
		No
		Yes

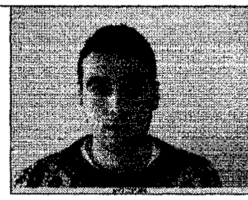
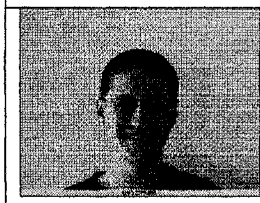
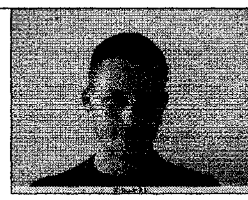
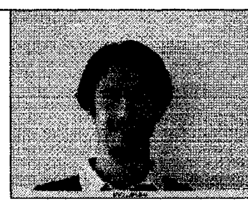
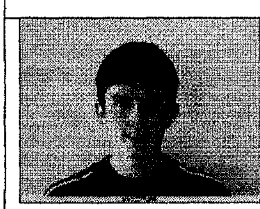
		Yes

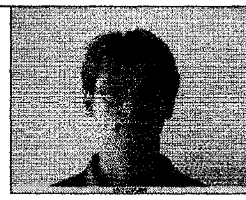
The recognition rate is 12/15 = 0.80

6.8 FLD Using color dataset

The results are shown in the following table:

Recognition Image	Recognized as	Recognized correctly?
		Yes
		Yes
	No recognition	No
		Yes

		Yes
		Yes
		Yes
		Yes
		Yes
		Yes
		Yes

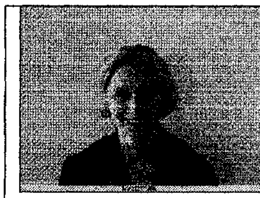
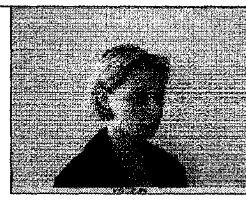
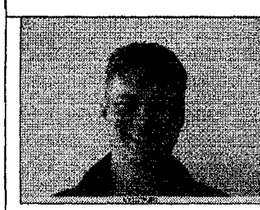
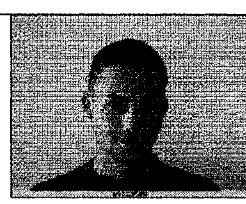
		Yes
		Yes
		Yes
		Yes

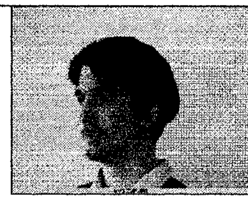
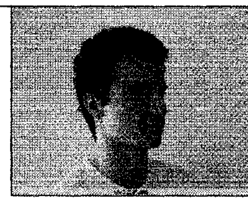
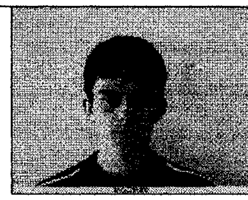
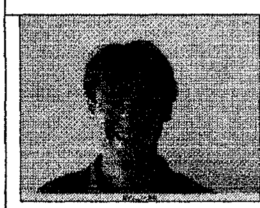
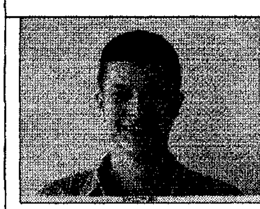
The recognition rate is 14/15 = 0.86

6.9 PCA using color dataset + Gabor filters

The results are shown in the following table:

Recognition Image	Recognized as	Recognized correctly?
		Yes

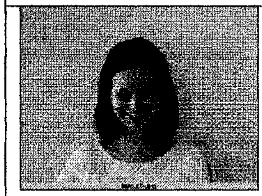
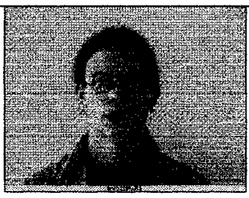
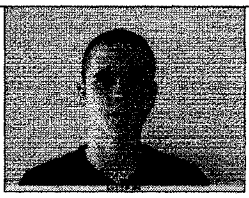
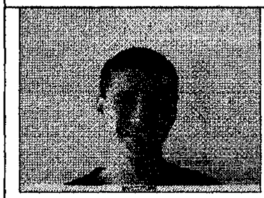
		Yes
		Yes
		Yes
		Yes
		No
		Yes
		Yes

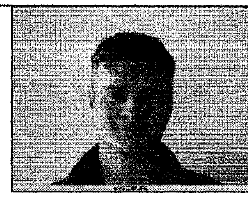
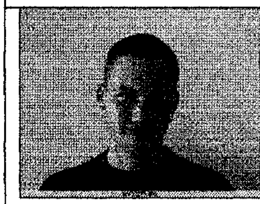
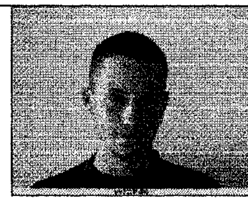
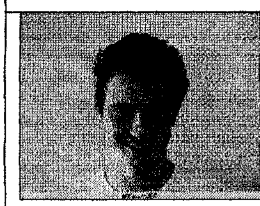
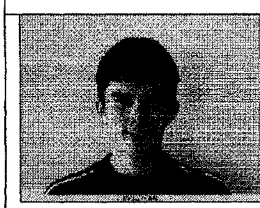
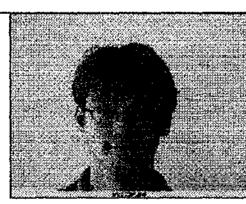
		Yes
		Yes
		Yes
		Yes
		Yes
		Yes
		Yes

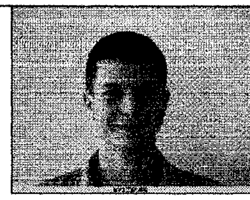
The recognition rate is 14/15 = 0.93

6.10 FLD using color dataset + Gabor filters

The results are shown in the following table:

Recognition Image	Recognized as	Recognized correctly?
		Yes
		Yes
		Yes
		Yes
		Yes
		No

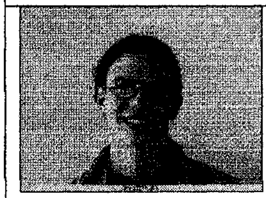
		Yes
		Yes
		Yes
		Yes
		Yes
		Yes
		Yes

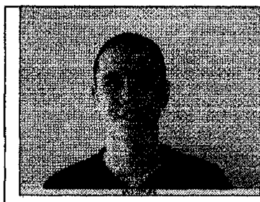
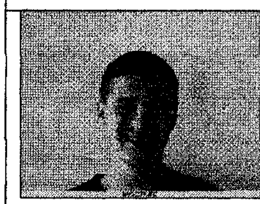
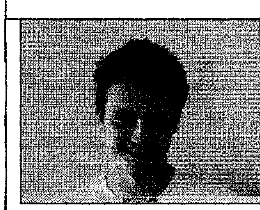
		Yes
		Yes

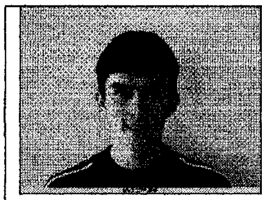
The recognition rate is 14/15 = 0.93

6.11 Discriminative Common Vectors using color dataset

The results are shown in the following table:

Recognition Image	Recognized correctly?
	Yes
	Yes
	Yes

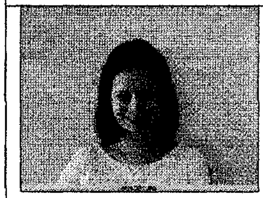
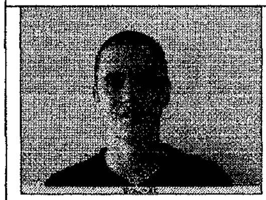
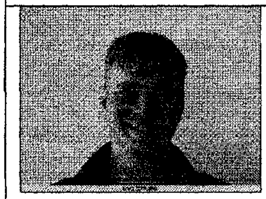
	Yes
	Yes
	Yes
	Yes
	Yes
	Yes
	Yes

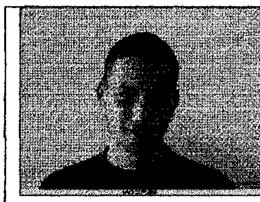
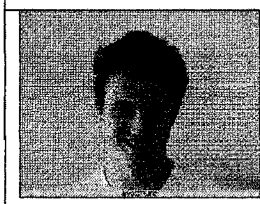
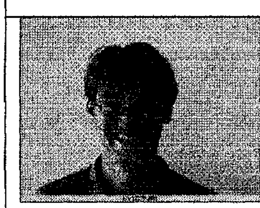
	Yes
	Yes
	Yes
	Yes
	Yes

The recognition rate is 15/15 = 1.00

6.12 *Laplacianfaces using color dataset*

The results are shown in the following table:

Recognition Image	Recognized correctly?
	Yes
	Yes
	No
	Yes
	Yes
	Yes
	Yes

	Yes
	Yes
	Yes
	Yes
	Yes
	Yes
	Yes

	Yes

The recognition rate is 14/15 = 0.93

6.13 Conclusion of experiments

The following table summarizes the result of the experiments:

	Gray-scaled dataset	Color dataset
PCA	0.73	0.80
FLD	0.80	0.93
Discriminative Common Vector	0.93	1.00
Laplacianfaces	0.93	0.93
PCA+Gabor filers	0.86	0.93
FLD+Gabor filters	0.86	0.93

Table 6.1 – Comparison for different Face Recognition algorithms

The recognition rate is increased when using FLD (compared to PCA) for gray-scaled dataset. The same is true for color dataset. Thus, FLD performs better on both gray-scaled and color datasets. Using Discriminative Common Vector increases the accuracy of recognition. For color images there was no error found and all the images were recognized correctly. Laplacianfaces does not have any advantage over Discriminative Common Vector method when using gray-scaled dataset. But Discriminative Common Vector has advantage over Laplacianfaces when using the color dataset.

Applying Gabor filters increased the accuracy of recognition. In this experiment, using the color dataset and applying Gabor filters, the recognition rate is increased compared to methods that do not use Gabor filters. The experiments were not done for the gray-scaled dataset since the focus of this study was on the role of color in face recognition. Fig. 6.1 is the conclusion of all the experiments that have been done using the dataset:

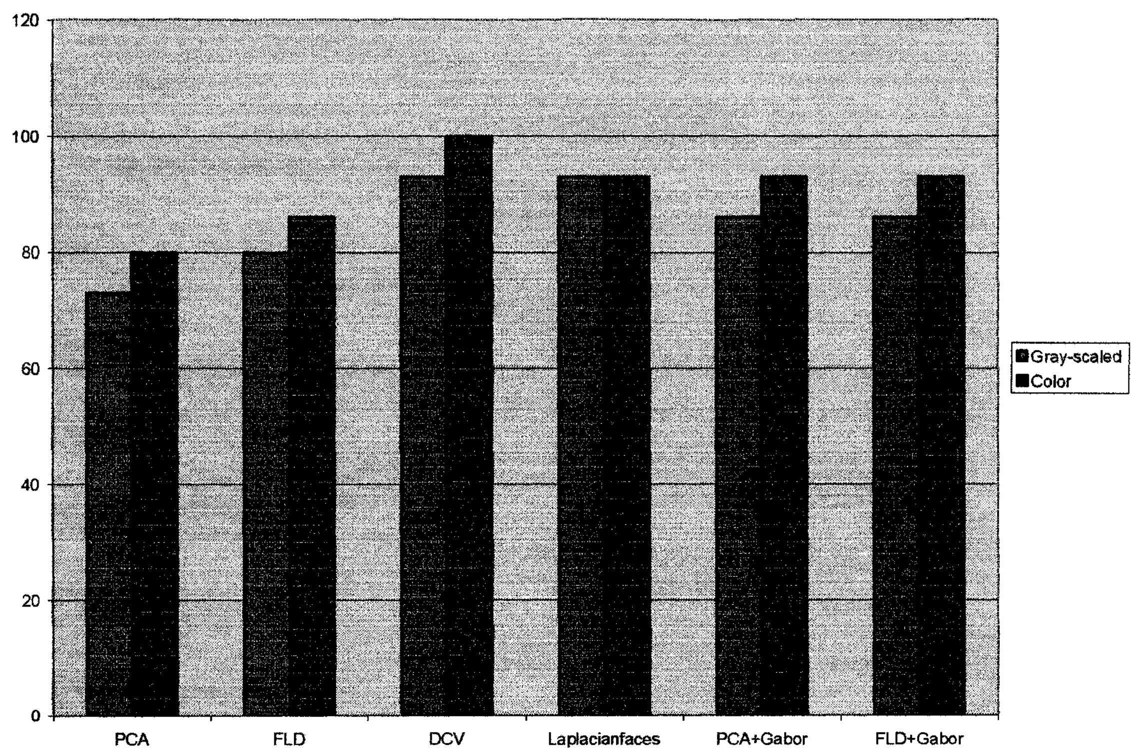


Fig. 6.1 – Recognition rates for different Face Recognition algorithms

6.14 Color Weights in the experiments

The study also looked at the weight of the colors in recognition, that is the study measured which color has more value in the recognition of color images. The following table illustrates this observation:

	Red	Green	Blue
PCA	0.83	0.33	0.75
FLD	0.84	0.38	0.69
Discriminative Common Vector	0.86	0.40	0.73
PCA+Gabor filers	0.86	0.36	0.71
FLD+Gabor filers	0.72	0.50	0.57

Table 6.2 – Weight of different colors in Face Recognition algorithms

Red had more weight, followed by Blue and Green has the least impact on recognition.

7. Experiments Using Georgia Tech Face Database

A color database (Georgia Tech Face Database) was selected for the experiments. The database contains images of 50 people and is stored in JPEG format. For each individual, there are 15 color images captured between 06/01/99 and 11/15/99. Most of the images were taken in two different sessions to take into account the variations in illumination conditions, facial expression, and appearance. In addition, the faces were captured at different scales and orientations.

From this database, 8 individuals have been selected and for each individual, 6 images per individual were chosen. In the training stage, the first 5 images for each individual have been selected. The last image was used for the recognition stage. Appendix III illustrates the selected set for training and recognition. There is 1 woman and 7 men.

This data set is used for different experiments. To understand the effect of colors in face recognition, the experiments have been done using color and gray-scaled images. Gray-scaled images have been constructed from the color images dataset. The formula in section 6 was used to convert color images into gray-scaled images. Four different methods for face recognition were used in these experiments: PCA, FLD, Discriminative Common Vectors and Laplacianfaces. Gabor filters were also applied to some of them to test recognition. Here is the list of experiments performed:

1) PCA using gray-scaled dataset
2) FLD using gray-scaled dataset
3) PCA using color dataset
4) FLD using color dataset
5) PCA using color dataset + Gabor filters

6) FLD using color dataset + Gabor filters

7) Discriminative Common Vectors using gray-scaled dataset

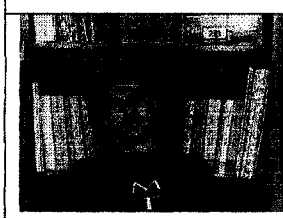
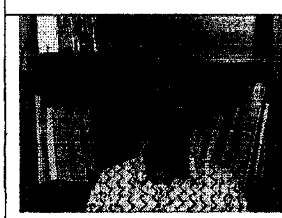
8) Discriminative Common Vectors using color dataset

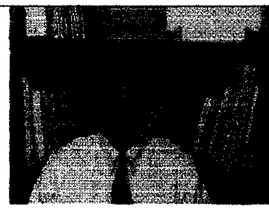
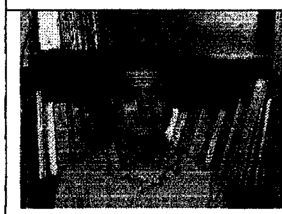
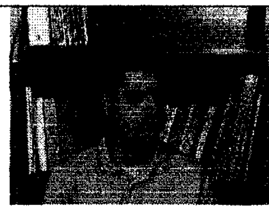
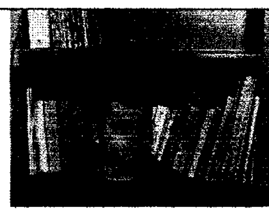
9) Laplacianfaces using gray-scaled dataset

10) Laplacianfaces using color dataset

7.1 PCA using gray-scaled dataset

The results are shown in the following table:

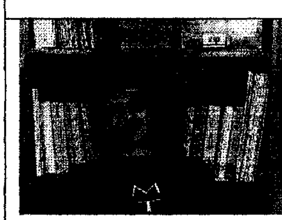
Recognition Image	Recognized as	Recognized correctly?
		No
		No
		Yes
		Yes

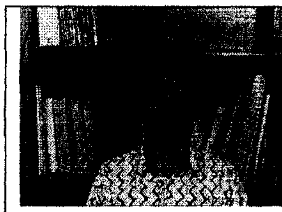
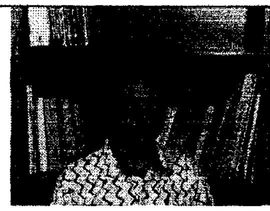
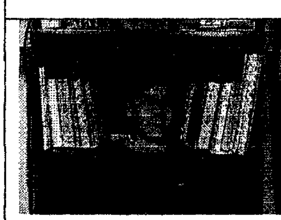
		Yes
		Yes
		Yes
		No

The recognition rate is 5/8 = 0.62

7.2 FLD using gray-scaled dataset

The results are shown in the following table:

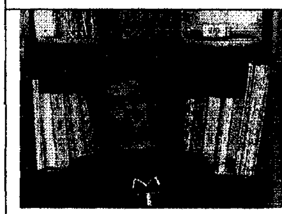
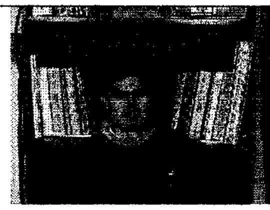
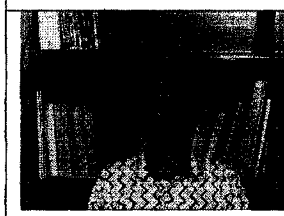
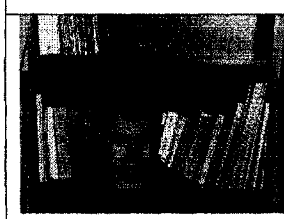
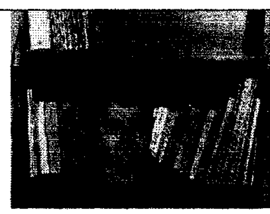
Recognition Image	Recognized as	Recognized correctly?
		Yes

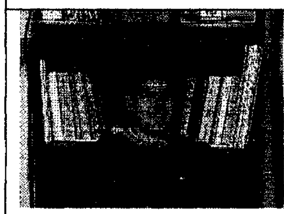
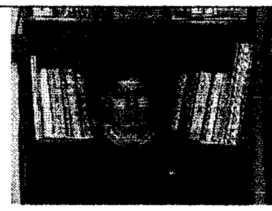
		Yes
		Yes
		Yes
		Yes
		Yes
		Yes
		No

The recognition rate is 7/8 = 0.87

7.3 PCA using gray-scaled dataset + Gabor Filters

The results are shown in the following table:

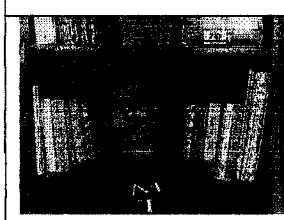
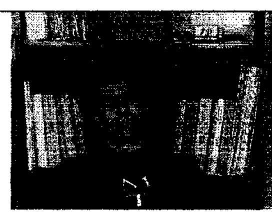
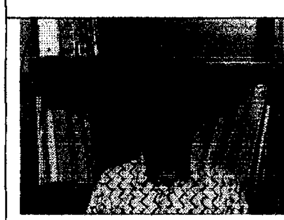
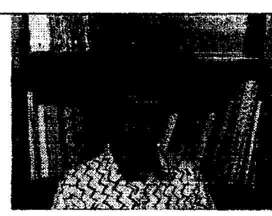
Recognition Image	Recognized as	Recognized correctly?
		No
		No
		Yes
		Yes
		Yes

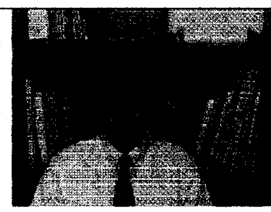
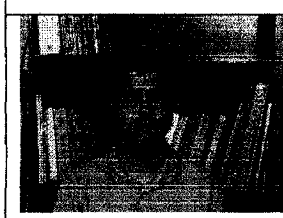
		Yes
		Yes
		Yes

The recognition rate is 6/8 = 0.75

7.4 FLD using gray-scaled dataset + Gabor Filters

The results are shown in the following table:

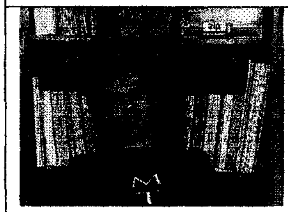
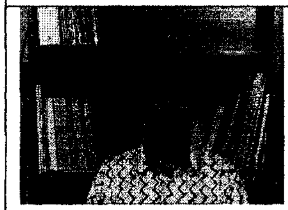
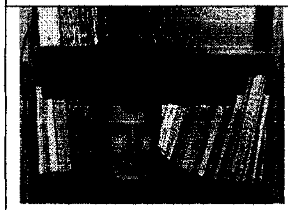
Recognition Image	Recognized as	Recognized correctly?
		Yes
		Yes

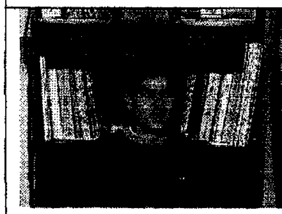
		Yes
		Yes
		Yes
		Yes
		Yes
		No

The recognition rate is 7/8 = 0.87

7.5 Discriminative Common Vectors using gray-scaled dataset

The results are shown in the following table:

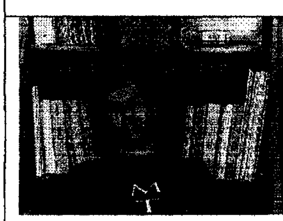
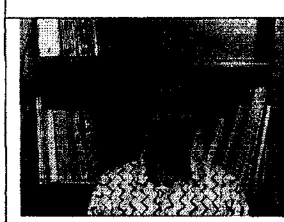
Recognition Image	Recognized correctly?
	Yes
	Yes
	Yes
	Yes
	Yes
	Yes

	Yes
	Yes

The recognition rate is 8/8 = 1.00

7.6 Laplacianfaces using gray-scaled dataset

The results are shown in the following table:

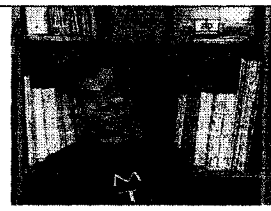
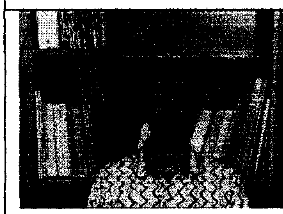
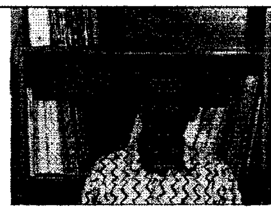
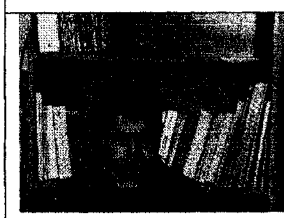
Recognition Image	Recognized correctly?
	Yes
	Yes
	Yes

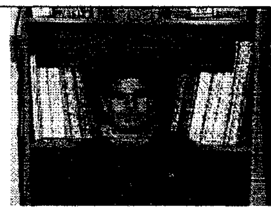
	Yes
	Yes
	Yes
	Yes
	Yes

The recognition rate is 8/8 = 1

7.7 PCA using color dataset

The results are shown in the following table:

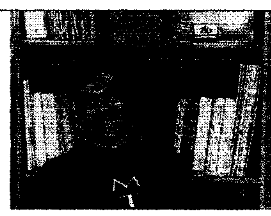
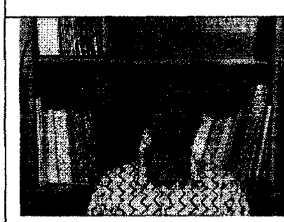
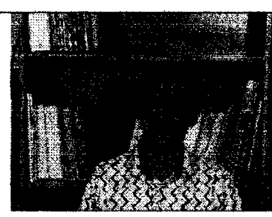
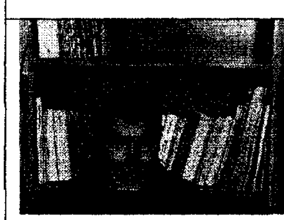
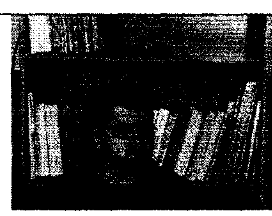
Recognition Image	Recognized as	Recognized correctly?
		Yes
		Yes
		Yes
		Yes
		Yes
		No

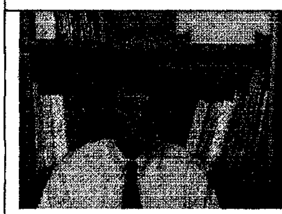
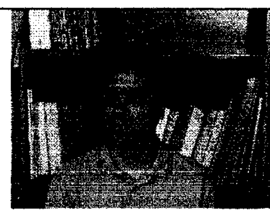
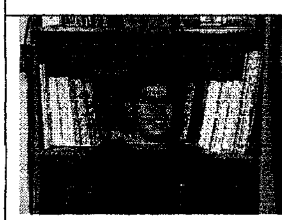
		Yes
		Yes

The recognition rate is 7/8 = 0.87

7.8 FLD Using color dataset

The results are shown in the following table:

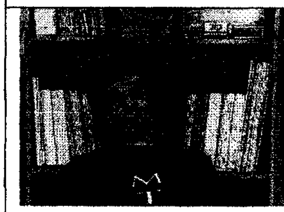
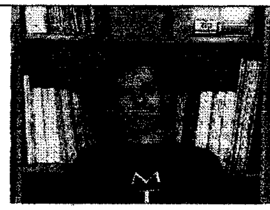
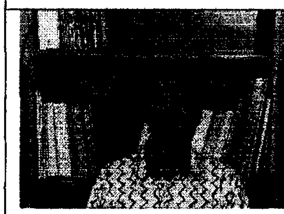
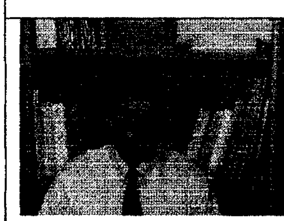
Recognition Image	Recognized as	Recognized correctly?
		Yes
		Yes
		Yes

		Yes
		Yes
		Yes
		Yes
	No Recognition	No

The recognition rate is 7/8 = 0.87

7.9 PCA using color dataset + Gabor filters

The results are shown in the following table:

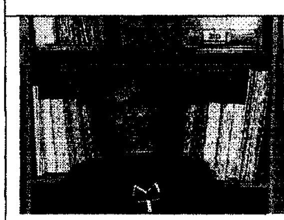
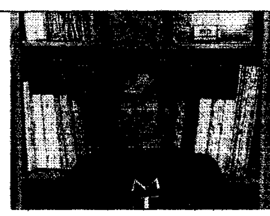
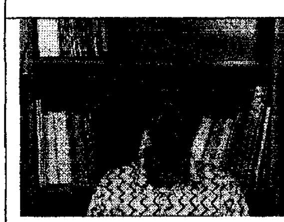
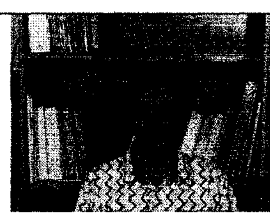
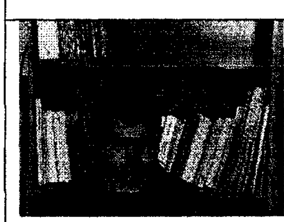
Recognition Image	Recognized as	Recognized correctly?
		Yes
		Yes
		Yes
		Yes
		Yes
		Yes

		Yes
		Yes

The recognition rate is 7/8 = 0.87

7.10 FLD using color dataset + Gabor filters

The results are shown in the following table:

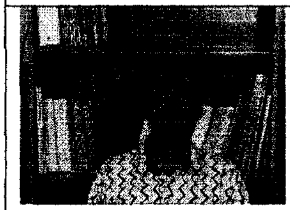
Recognition Image	Recognized as	Recognized correctly?
		Yes
		Yes
		Yes

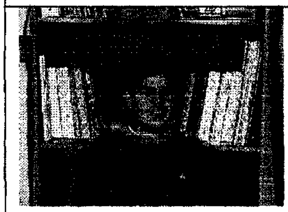
		Yes
		Yes
		Yes
		Yes
		Yes

The recognition rate is 8/8 = 1.00

7.11 Discriminative Common Vectors using color dataset

The results are shown in the following table:

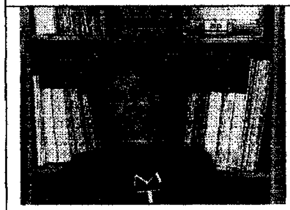
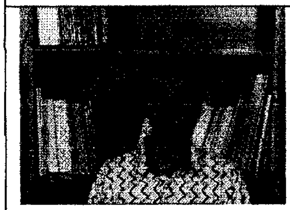
Recognition Image	Recognized correctly?
	Yes
	Yes
	Yes
	Yes
	Yes
	Yes

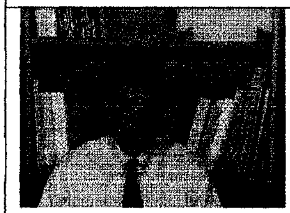
	Yes
	Yes

The recognition rate is 8/8 = 1.00

7.12 Laplacianfaces using color dataset

The results are shown in the following table:

Recognition Image	Recognized correctly?
	Yes
	Yes
	Yes

	Yes
	Yes
	Yes
	Yes
	Yes

The recognition rate is 8/8 = 1.00

7.13 Conclusion of experiments

The following table summarizes the result of our experiments:

	Gray-scaled dataset	Color dataset
PCA	0.62	0.87
FLD	0.87	0.87
Discriminative Common Vector	1.00	1.00
Laplacianfaces	1.00	1.00
PCA+Gabor filers	0.75	0.87
FLD+Gabor filters	0.87	1.00

Table 7.1 – Comparison for different Face Recognition algorithms

The recognition rate is increased when using FLD (compared to PCA) for the gray-scaled dataset. In this experiment, color did not increase the recognition rate for FLD. Using Discriminative Common Vector increases the accuracy of recognition. For color images there was no error found and all the images were recognized correctly. Laplacianfaces does not have any advantage over Discriminative Common Vector method when using the gray-scaled dataset. Applying Gabor filters also increased the accuracy of recognition for PCA. Fig. 7.1 is the conclusion of all the experiments that have been done using the mentioned dataset:

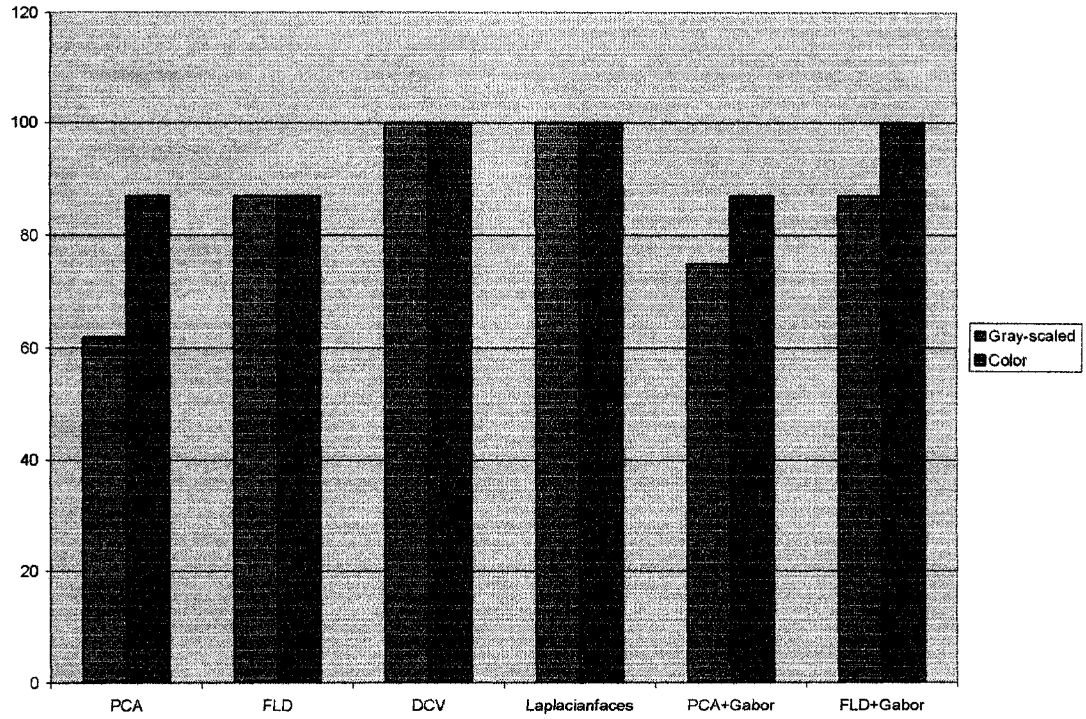


Fig. 7.1 – Recognition rates for different Face Recognition algorithms

7.14 Color Weights in the experiments

In this experiment the weight of colors in recognition was measured. The following table illustrates this observation:

	Red	Green	Blue
PCA	1.00	0.50	0.66
FLD	1.00	0.71	0.71
Discriminative Common Vector	0.87	0.75	0.75
PCA+Gabor filers	0.87	0.75	0.87
FLD+Gabor filers	0.87	0.75	0.87

Table 7.2 – Weight of different colors in Face Recognition algorithms

Red had the highest recognition, Blue was second and Green had the least impact on recognition.

8. Implementation Notes

Implementation of the application has been done using Microsoft.NET. The language that has been used is C#, which is an object oriented language. In the implementation, I tried to use object oriented concepts in order to make different algorithms more reusable.

8.1 Linear Algebra Operations

A C# library is used for basic linear algebra operations. The classes in this library give a basic linear algebra package for .NET. It provides user-level C# classes for constructing and manipulating real, dense matrices. It is meant to provide sufficient functionality for routine problems, packaged in a way that is natural and understandable to non-experts. That said, it is a port of a public domain Java matrix library, called JAMA.

8.1.1 Background (From library Developer)

Currently, the developer of this library (Paul Selormey) works for a small GIS company in Japan developing GIS components for application developers. Coordinate Transformation and therefore Affine Transformation is a very basic part of the development efforts. Recently, he was assigned a task of designing and implementing a new GIS system for the .NET framework with the ability to easily port to Java and other frameworks. He decided to make maximum use of matrix-based affine transformation, which is also a requirement of the OpenGIS Coordinate Transformation Specifications.

Then, he discovered that the Matrix class provided as part of the GDI+ in the .NET implements the affine transformations in a manner different from standard specifications. In short, while the standard 2D Coordinate System Affine Transformation Matrix is define as

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

The matrix defined by GDI+ is

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{pmatrix}$$

The effect is that most affine transformations with the GDI+ Matrix class will not conform to standard or specifications. For instance, by standard (and mathematically) anti-clockwise (or counter-clockwise) rotations are considered positive but must be negative when using the GDI+ classes. To solve this problem, he decided to implement a standard affine transformation matrix and found this small Java general matrix library - JAMA. The presented classes are ported from the JAMA with .NET specific improvements like operator overloading etc. The library is referred to as DotNetMatrix, which provides linear algebra operations.

8.1.2 Using the library

DotNetMatrix is comprised of six C# classes: GeneralMatrix, CholeskyDecomposition, LUDecomposition, QRDecomposition, SingularValueDecomposition and EigenvalueDecomposition. The GeneralMatrix class provides the fundamental operations of numerical linear algebra. Various constructors create Matrices from two dimensional arrays of double precision floating point numbers. Various gets and sets (properties) provide access to submatrices and matrix elements. The basic arithmetic operations include matrix addition and multiplication, matrix norms and selected element-by-element array operations. A convenient matrix print method is also included.

Five fundamental matrix decompositions, which consist of pairs or triples of matrices, permutation vectors, and the like, produce results in five decomposition classes. These decompositions are accessed by the GeneralMatrix class to compute solutions of simultaneous linear equations, determinants, inverses and other matrix functions. The five decompositions are

- Cholesky Decomposition of symmetric, positive definite matrices
- LU Decomposition (Gaussian elimination) of rectangular matrices
- QR Decomposition of rectangular matrices
- Eigenvalue Decomposition of both symmetric and nonsymmetric square matrices
- Singular Value Decomposition of rectangular matrices

- The DotNetMatrix deals only with real matrices, there is not support for complex matrices.

The design of DotNetMatrix represents a compromise between the need for pure and elegant object-oriented design and the need to enable high performance implementations. Table 7.1 illustrates a summary of DotNetMatrix library capabilities:

Object Manipulation	constructors set elements get elements copy clone
Elementary Operations	addition subtraction multiplication scalar multiplication element-wise multiplication element-wise division unary minus transpose norm
Decompositions	Cholesky LU QR SVD symmetric eigenvalue nonsymmetric eigenvalue
Equation Solution	nonsingular systems least squares
Derived Quantities	Condition number determinant rank inverse pseudoinverse

Table 7.1 - Summary of DotNetMatrix Capabilities

8.2 *Implementation of PCA*

8.2.1 Training Stage

In the training stage of PCA algorithm, the following steps were performed:

- Loading training images into an array
- Calculating Mean image for the loaded images
- Subtracting mean image from all the images in the training set
- Constructing eigenfaces

The steps mentioned are achieved using the following code:

```
// Load the pictures into the memory.
this.LoadPics();
// Mean Matrix is being calculated
MeanMatrix = (new MiddleTier.Extraction()).ReturnMeanMatrix(X);
A = (new MiddleTier.Extraction()).CalculateMean(X);
U = MiddleTier.PCA.TrainPCA(X);
```

LoadPics method loads the training set into a global array (X), MeanMatrix is the mean image for all the images in the training set, A is the matrix constructed by subtracting mean image from all the images in the training set and U is the eigenfaces. All the values are scaled in the matrix because the values for each pixel in U matrix might not be in the range of 0-255. It is achieved by using the following code:

```
ScaledU = GeneralMatrixHelper.ScaleMatrixToGrayScale(U);
```

In which ScaledU is a global variable to hold the scaled eigenfaces matrix.

8.2.2 Recognition Stage

In the recognition stage, calculated matrixes from the previous stage are passed into a recognition routine:

```
MiddleTier.PCA.RecognizePCA(U,A,MeanMatrix,RecognitionPic);
```

This routing receives the recognition picture and the required parameters (calculated in the Training Stage) and performs nearest neighbor algorithm as explained in section 3.1.3.

8.3 Implementation of FLD

8.3.1 Training Stage

In the training stage of FLD algorithm, the following steps were performed:

- Loading training images into an array
- Calculating Mean image for the loaded images
- Calculating Class mean for each class
- Calculating OrthonormalImage and Orthonormal Mean matrixes
- Calculating scatter matrices
- Constructing eigenfaces

The steps mentioned are achieved using the following code:

```
// first we loop through each class and store the class in the memory
for (int i=0; i<this.NumberOfPersons; i++)
{
      GeneralMatrix Temp = new GeneralMatrix(this.Width*this.Height,
      this.NumberOfPicPerPerson);
      for (int j=0; j<this.NumberOfPicPerPerson; j++)
      {
            for (int k=0; k<this.Width*this.Height; k++)
            {
                  Temp.SetElement(k,j, X.GetElement(k,i+j));
            }
      }
      // Claculate Mean for each class
      GeneralMatrix Mean = Ex.ReturnMeanMatrix(Temp);
      // Now center each image in each class
      for (int m=0; m<this.NumberOfPicPerPerson; m++)
      {
            for (int n=0; n<this.Width*this.Height; n++)
            {
                  ImageMean.SetElement(n, m+i, X.GetElement(n,m+i) -
                  Mean.GetElement(n,0));
            }
      }
      // calculate Class Mean
      for (int k=0; k<Mean.RowDimension; k++)
```

```
        {
                Mu_i.SetElement(k,i, Mean.GetElement(k,0));
                ClassMean.SetElement(k,i, Mu_i.GetElement(k,0) -
                MeanMatrix.GetElement(k,0));
        }
}

// calculate OrthonormalImage and Orthonormal
Mu = Ex.ReturnMeanMatrix(Mu_i);
QRDecomposition QR = new QRDecomposition(X);
GeneralMatrix W = QR.Q;
GeneralMatrix OrthonormalImage = W.Transpose().Multiply(ImageMean);
GeneralMatrix OrthonormalMean = W.Transpose().Multiply(ClassMean);
// calculate scatter matrices
GeneralMatrix S_w =
OrthonormalImage.Multiply(OrthonormalImage.Transpose());

GeneralMatrix S_b = new GeneralMatrix(OrthonormalMean.RowDimension,
OrthonormalMean.RowDimension);
GeneralMatrix Tmp;

GeneralMatrix S = S_w.Inverse().Multiply(S_b);

// construct eigenfaces
EigenvalueDecomposition ED = new EigenvalueDecomposition(S);
Ufld = X.Multiply(ED.GetV());
```

The LoadPics method loads the training set into a global array (X). Then ImageMean and ClassMean matrixes are calculated, which represent the mean of all the images and also the mean of all the images in one specific class. Scattered matrixes are also calculated (S_w and S_b). Finally, ED, which is the Eigen value decomposition, is calculated based on the scattered matrixes and the U matrix for FLD is generated.

8.4 Implementation of Discriminative Common Vectors

8.4.1 Training Stage

In the training stage of Discriminative Common Vectors algorithm, the following steps were performed:

- Loading training images into an array
- Calculating Mean image for the loaded images
- Calculating Class mean for each class
- Calculating projection into null space (matrix Q)
- Calculating common vectors for each class
- Construct the weight matrix to be used in recognition stage

The steps mentioned are achieved using the following code:

```
// calculating mean matrix
MeanMatrix = (new MiddleTier.Extraction()).ReturnMeanMatrix(X);
A = (new MiddleTier.Extraction()).CalculateMean(X);
```

```
// calculating class mean
for (int k=0; k<Mean.RowDimension; k++)
{
      Mu_i.SetElement(k,i, Mean.GetElement(k,0));
      ClassMean.SetElement(k,i,          Mu_i.GetElement(k,0)          -
      MeanMatrix.GetElement(k,0));
}

// calculating Q matrix
EigenvalueDecomposition ED = new EigenvalueDecomposition(S_w);
GeneralMatrix Q = A.Multiply(ED.GetV());

// calculating Common Vectors for each class
GeneralMatrix Temp1 = Q.Multiply(Q.Transpose());
XComm = XTemp.Subtract(Temp1.Multiply(XTemp));
AComm = (new MiddleTier.Extraction()).CalculateMean(XComm);

// weight matrix for recognition stage
U = MiddleTier.PCA.TrainPCA(XComm);
```

8.4.2 Recognition Stage

In the recognition stage, calculated matrixes from the previous stage are passed into a recognition routine. The recognition stage in this algorithm is very similar to PCA and will not be explained in detail. This routing receives the recognition picture and the required parameters (calculated in the Training Stage) and performs the nearest neighbor algorithm as explained in section 3.2

8.5 *Implementation of Laplacianfaces*

8.5.1 Training Stage

In the training stage of Laplacianfaces algorithm, the following steps were performed:

- Calculate Similarity matrix
- Calculate Diagonal matrix D, explained in section 3.5.1
- Calculate Laplacian Matrix L, explained in section 3.5.1
- Calculate Transformation Vector

The steps mentioned are achieved using the following code:

```
// calculate Similarity matrix

for (int i=0; i<this.NumberOfPersons*this.NumberOfPicPerPerson; i++)
{
	for (int j=0; j<this.NumberOfPersons*this.NumberOfPicPerPerson; j++)
	{
		if (i!=j)
		{
			// extract vector
			for (int k=0; k<this.Width*this.Height; k++)
			{
				TempXi.SetElement(k,0,X.GetElement(k,i));
				TempXj.SetElement(k,0,X.GetElement(k,j));
			}
			double t1 = Math.Exp(-x.GetElement(0,0)/1000);
			S.SetElement(i,j,t1);
		}
	}
}

// calculate D matrix

for (int i=0; i<this.NumberOfPersons*this.NumberOfPicPerPerson; i++)
{
	double tmp = 0;
	for (int j=0; j<this.NumberOfPersons*this.NumberOfPicPerPerson; j++)
	{
		tmp = tmp + S.GetElement(i,j);
	}
```

```
        D.SetElement(i,i,tmp);
}

// calculate Laplacian Matrix

GeneralMatrix L = D.Subtract(S);

// calculate Transformation matrix

U = X.Multiply(L);
```

8.5.2 Recognition Stage

The transformation matrix explained in section 3.5.1 using LLP method is used in this stage for recognition. This matrix plays the role of eigenfaces in PCA method. The same approach for PCA is used in the recognition using the shortest distance.

9. Conclusions

Several face recognition algorithms were introduced in recent years. The new algorithms promised to have better recognition rate. The goal for researchers was to try to implement recognition algorithms that are more accurate in terms of face identification. This study implemented and compared five face recognition algorithms and ran them against a dataset. Originally, the dataset contained color images but they were converted into grayscale images so that experiments could be done on both grayscale and color datasets.

This research indicates that color is an important cue for face recognition. For traditional face recognition methods like PCA and FLD, which do not provide a high recognition rate, using color images improves the recognition accuracy significantly. Even for the algorithms, where recognition rates are near the peak, there are still some improvements obtained.

Findings also indicate that the color Red has more weight than other colors in face recognition. This may also explain the debate surrounding the role of color in face recognition. It has been speculated in the literature [36] that color is not a very important cue in face recognition because color-blind people can identify the faces at the same rate as other non color-blind people. In light of our research such a statement seems unfounded. Considering that color-blind people have difficulty distinguishing between Red and Green, it is likely that high weights for color Red and Blue compensate for color Green and recognition rate stays the same for color-blind people. Further research is needed to prove this hypothesis.

10. Summary of Contributions

Here is the summary of contributions for this thesis:

- Several face recognition algorithms including two new face recognition algorithms were implemented (Laplacianfaces and Discriminative Common Vector)
- A technique for recognition of color images was implemented
- Several experiments were done using traditional and new face recognition algorithms
- The role of color in face recognition was identified
- Weight of each color has been identified to see which colors are more important in face recognition
- The argument about color-blind people stating that "Color blind people recognize images at the same rate as non-color-blind people, so color is not an important cue in face recognition" was discussed.

References

[1] A. Abadpour, S. Kasaei, "New PCA-based compression method for natural color images", *International Workshop on Computer Vision,* April 26-30, IPM, Tehran, 2004

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

[3] A. Yip and P. Sinha, "Role of color in face recognition", *Journal of Vision*, vol. 2, no. 7, p. 596a, 2001

[4] C. BenAbdelkader, P. Griffin, "A local region-based approach to gender classification from face images," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),* Workshops, pp. 52-52, 2005

[5] C. Chen, S.P. Chiang, "Detection of human faces in color images", *VISP,* vol. 144, no. 6, pp. 384-388, 1997

[6] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced Fisher Linear Discriminant model for face recognition", *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 467-476, 2002

[7] C. Liu, H. Wechsler, "Comparative assessment of independent component analysis (ICA) for face recognition", *Proc. of the Second International Conference on Audio- and Video-based Biometric Person Authentication*, pp. 211-216, 1999

[8] D. Pissarenko. "Eigenface-based facial recognition", Feb. 2002

[9] D. Skoaj, H. Bischof, A. Leonardis, "A robust PCA algorithm for building representations from panoramic images", *ECCV,* vol. 2, p. 761, 2002

[10] G. Fritz, L. Paletta, H. Bischof, "Object recognition using local information content", *ICPR04*, vol. 2, pp. 15-18, 2004

[11] G. Givens, J.R. Beveridge, B.A. Draper, D. Bolme, "A statistical assessment of subject factors in the PCA recognition of human subjects", *Proceedings of CVPR Workshop: Statistical Analysis in Computer Vision*, 2003

[12] G. Tambasis, "History of neural networks", (submitted for history of computation) University of Wales, Swansea, 1999

[13] H. Cevikalp, "Discriminative Commom Vectors for face recognition", IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 4-13, Jan. 2005

[14] H. Moon, P.J. Phillips, "Computational and performance aspects of PCA-based face recognition algorithms", *Perception*, vol. 30, pp. 303-321, 2001

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

[16] I, Atalay. "Face recognition". Master Thesis, Istanbul Technical University, 1996

[17] I. Biederman, "Recognition-by-Components: A theory of human image understanding", *Psychological Review*, vol. 94, pp. 115--147, 1987

[18] J. C. Goswami, and A. K. Chan, Fundamentals of wavelets: theory, algorithms, and applications, Wiley, New York, 1999

[19] J. Edward Jackson, "A user's guide to principal components", *Wiley Series in Probability and Statistics*, 1991

[20] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, "Face recognition using LDA-based algorithms", *IEEE Trans. on Neural Networks*, vol. 14, no. 1, pp. 195-200, 2003

[21] K. Delac, M. Grgic, S. Grgic, "Independent comparative study of PCA, ICA, and LDA on the FERET data set", Technical Report, FER-VCL-TR-2004-03, 2004

[22] K. Kim, "Face recognition using principle component analysis", Tutorial, University of Maryland, College Park,

[23] L. I. Smith, "A tutorial on principal components analysis", December 2005, www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

[24] L. Torres, J. Y. Reutter and L. Lorente, "Importance of the color information in face recognition", Polytechnic University of Catalonia, Barcelona, Spain, 1999

[25] L. Zhang, D. Samaras, D. Tomasi, N. Volkow, R. Goldstein, "Machine learning for clinical diagnosis from functional magnetic resonance imaging," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 1211-1217, 2005

[26] M. Alwawi and H. Demirel, "PCA and LDA based face recognition under changing class sizes and non-uniform illumination conditions", *Proceedings of the Signal and Image Processing*, 2005

[27] M.A. Turk and A.P. Pentland. "Face recognition using Eigenfaces", *IEEE Conference on Computer Vision and Pattern Recognition*, pp.586--591, 1991

[28] M. Frazier, A*n introduction to wavelets through linear algebra*, Springer, New York, 1999

[29] M.J. Swain and D.H. Ballard, "Color indexing", *International Journal of Computing Vision*, vol. 7, no. 1, pp. 11-32, 1991.

[30] M. Petrou and J. Wiley, *Image processing: the fundamentals*, Chichester, New York, 1999

[31] M.S. Bartlett, J.R. Movellan, T.J. Sejnowski, "Face recognition by Independent Component Analysis", *IEEE Trans. on Neural Networks*, vol. 13, no. 6, pp. 1450-1464, 2002

[32] M. Savvides, B.V.K. Vijaya Kumar, P.K. Khosla, ""Corefaces" - robust shift invariant PCA based correlation filter for illumination tolerant face recognition," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04),* vol. 2, 2004

[33] M. Turk and A. Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1993

[34] O. O. Khalifa, M. Yusuf, and S. Gagula, "Face recognition using PCA wavelet decomposition", Electrical & Computer Department, Kuliyyah of Engineering, International Islamic University Malaysia, 2001

[35] P. Belhumeur, J. Hespanha, and D. Kriegman, “Eigenfaces vs. Fisherfaces: recognition using class specific linear projection”, In *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711-720, July 1997

[36] P. Cavanagh and Y. G. Leclerc, “Shape from shadows”, *Journal of Experimental Psychology: Human Perception and Performance*, vol. 15, no. 1, pp. 3-27, 1989

[37] P. Jonathon Phillips, Patrick J. Flynn, Todd Scruggs, Kevin W. Bowyer, Jin Chang, Kevin Hoffman, Joe Marques, Jaesik Min, William Worek, "Overview of the face recognition grand challenge", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 947-954, 2005

[38] P.J. Phillips, E.M. Newton, “Meta-Analysis of face recognition algorithms”, *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 235-241, 2002

[39] P. Navarrete, and J. Ruiz-del-Solar, “Comparative study between different Eigenspace-based approaches for face recognition”, *Lecture Notes in Computer Science*, vol. 2275, pp. 178-184, 2001

[40] P. N. Belhumeur, J. P. Hespanha, D. J. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition using class specific linear”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vo. 19, no. 7, pp. 711-720, 1997

[41] R. Gottumukkal, V. K. Asari, "Real time face detection from color video stream based on PCA method", *32nd Applied Imagery Pattern Recognition Workshop (AIPR'03),* p. 146, 2003.

[42] R. Kemp, G. Pike and A. Musselman, “Perception and recognition of normal and negative faces: The role of shape from shading and pigmentation cues”, *Perception*, vol. 25, pp. 37-52.

[43] R. Tjahyadi, W. Liu, S. An and S. Venkatesh, “Face recognition based on ordinal correlation approach”, *International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP2005)*, pp. 349-354, Melbourne, Australia, 2005

[44] S.G. Narasimhan, S.K. Nayar, B. Sun, and S.J. Koppal, "Structured light in scattering media," *International Conference on Computer Vision (ICCV)*, 2005

[45] S. Lin, H.Y. Shum, “Separation of diffuse and specular reflection in color images”, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 341-346, 2001

[46] T. Ho-Man, “Face recognition committee machine: Methodology, experiments and a system application”. Master Thesis, The Chinese University of Hong Kong, July 2003

[47] V. Bruce, and A. YOUNG, *In the eye of the beholder: the science of face perception*, Oxford University Press, 1998

[48] W. E. L. Grimson, *Object recognition by computer: the role of geometric constraints*, The MIT Press, Cambridge, Mass., 1990.

[49] W. Zhao, R. Chellappa, and A. Krishnaswamv, “Discriminant analysis of Principal Components for face recognition”, *Proc. Third IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 336-341, Apr. 1998

[50] W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips, "Face recognition: A literature survey", *ACM Computing Surveys*, vol. 35, no. 4, pp. 399-458, 2003

[51] W. Zhao, R. Chellappa, A. Krishnaswamy, "Discriminant analysis of Principal Components for face recognition", *Proc. of the 3rd IEEE International Conference on Face and Gesture Recognition*, p. 336, 1998

[52] X. He, S. Yan, Y. Hu, P. Niyogi, and H.J. Zhang, "Face recognition using Laplacianfaces", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328-340, March 2005

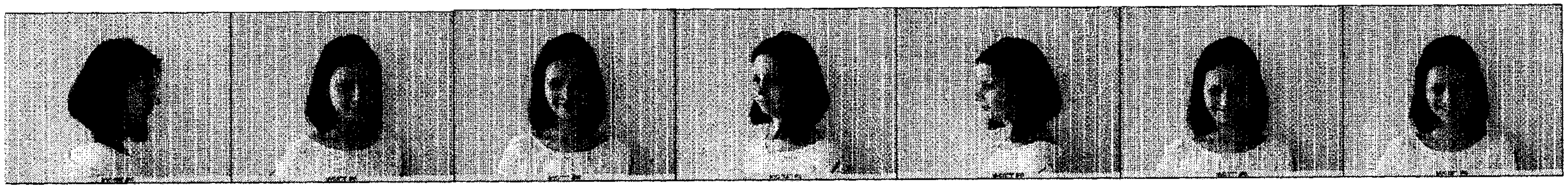
[53] X. Wang, X. Tang, "Subspace analysis using random mixture models", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 574 – 580, 2005.

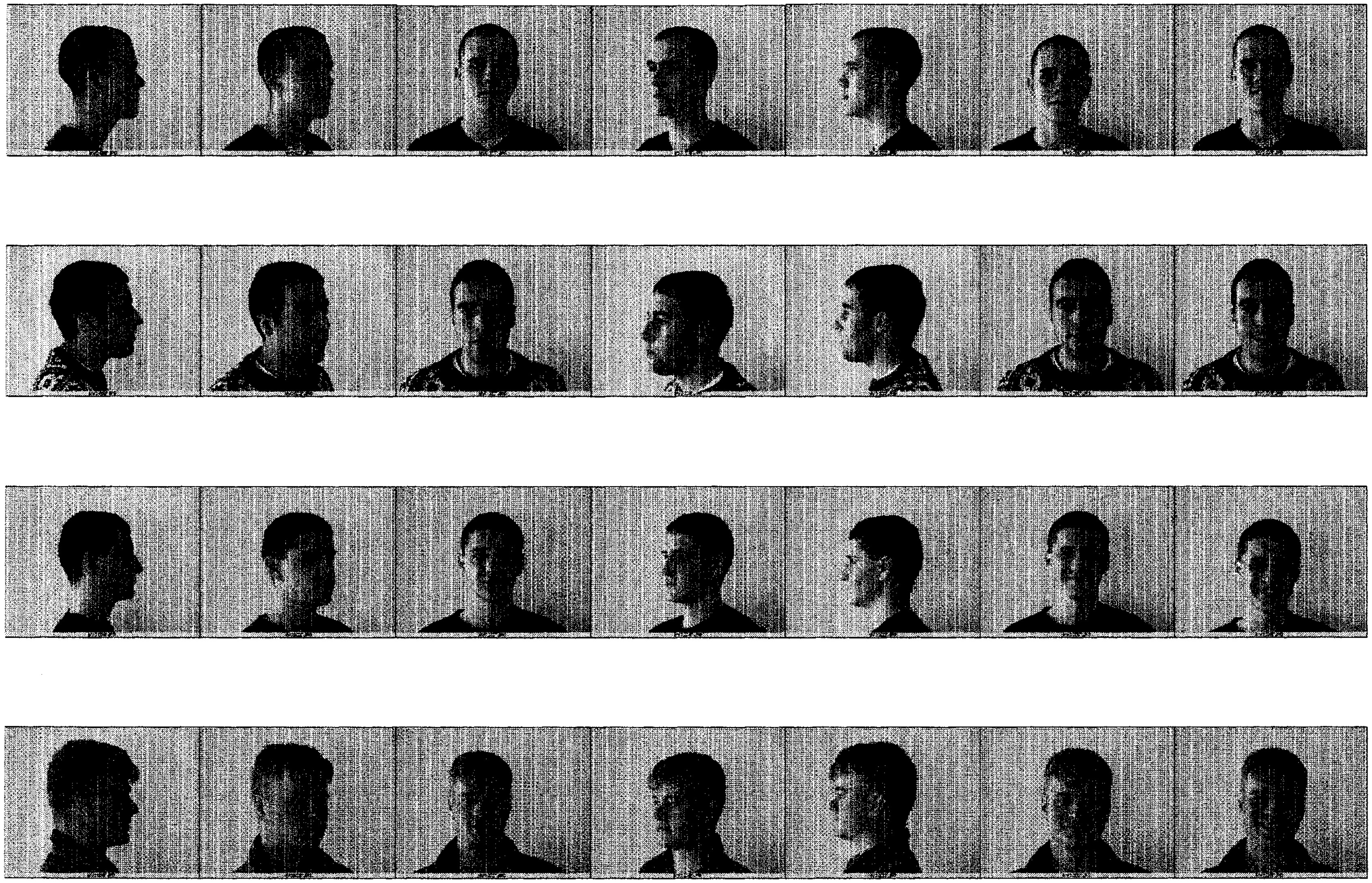
[54] Yambor, Wendy. "Analysis of PCA-based and Fisher discriminant-based image recognition algorithms". Master Thesis, Colorado State University, July 2000

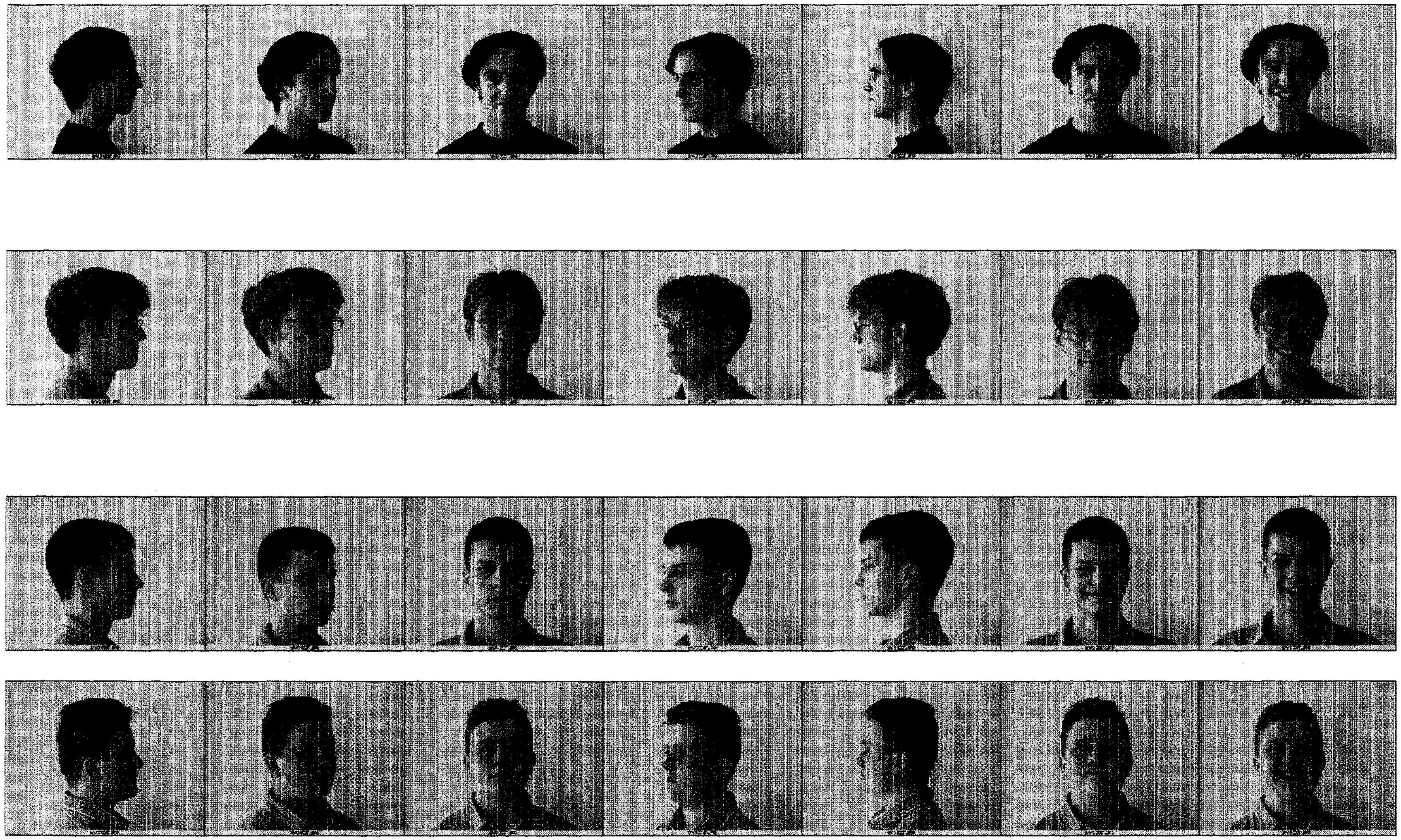
[55] Y. Lee, "Face Detection and recognition using linear classification", Master Thesis, University of Calgary, 2004

[56] "Principal component analysis", December 2005, http://en.wikipedia.org/wiki/Principal_components_analysis

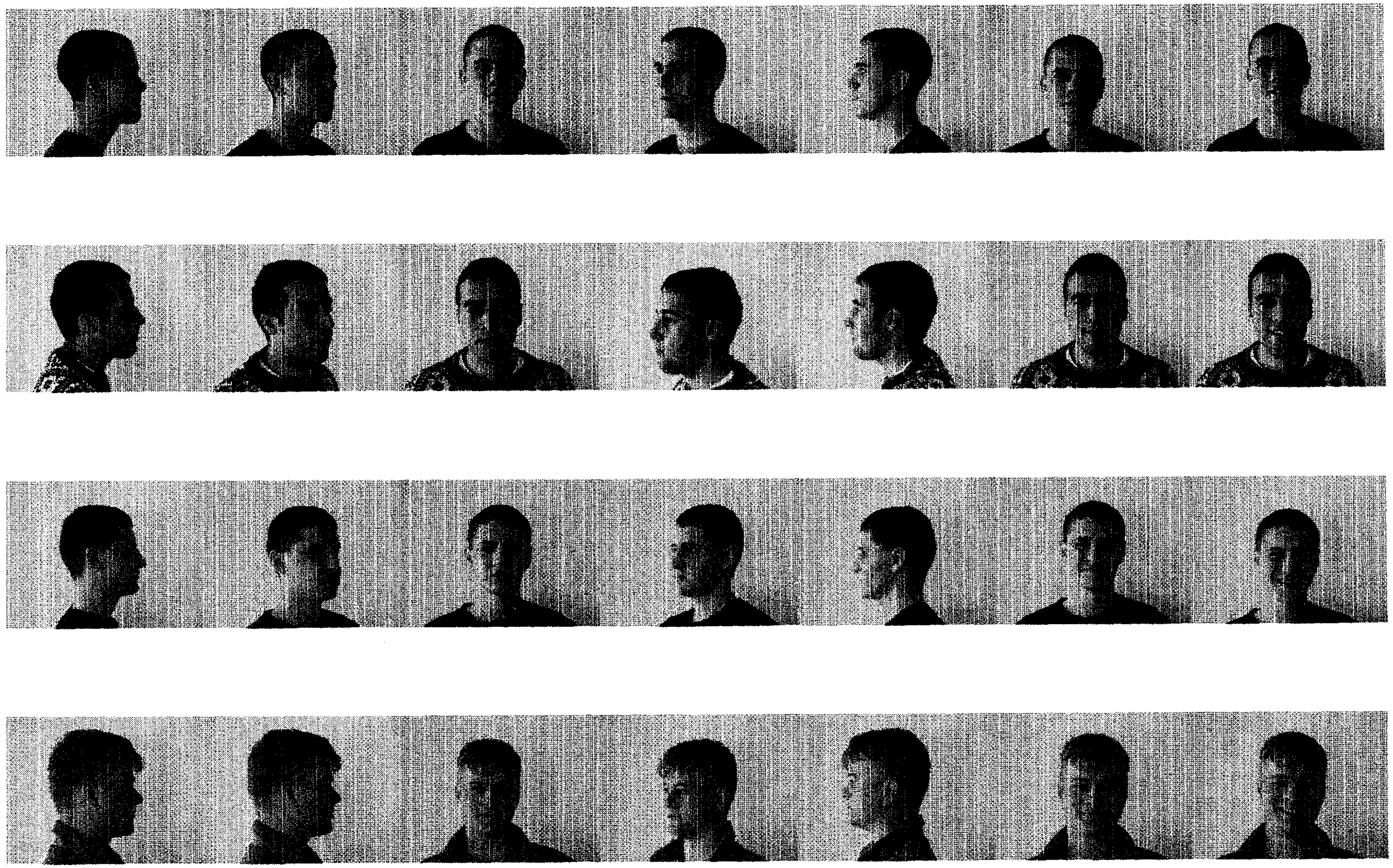
Appendix I – CVL Color Dataset

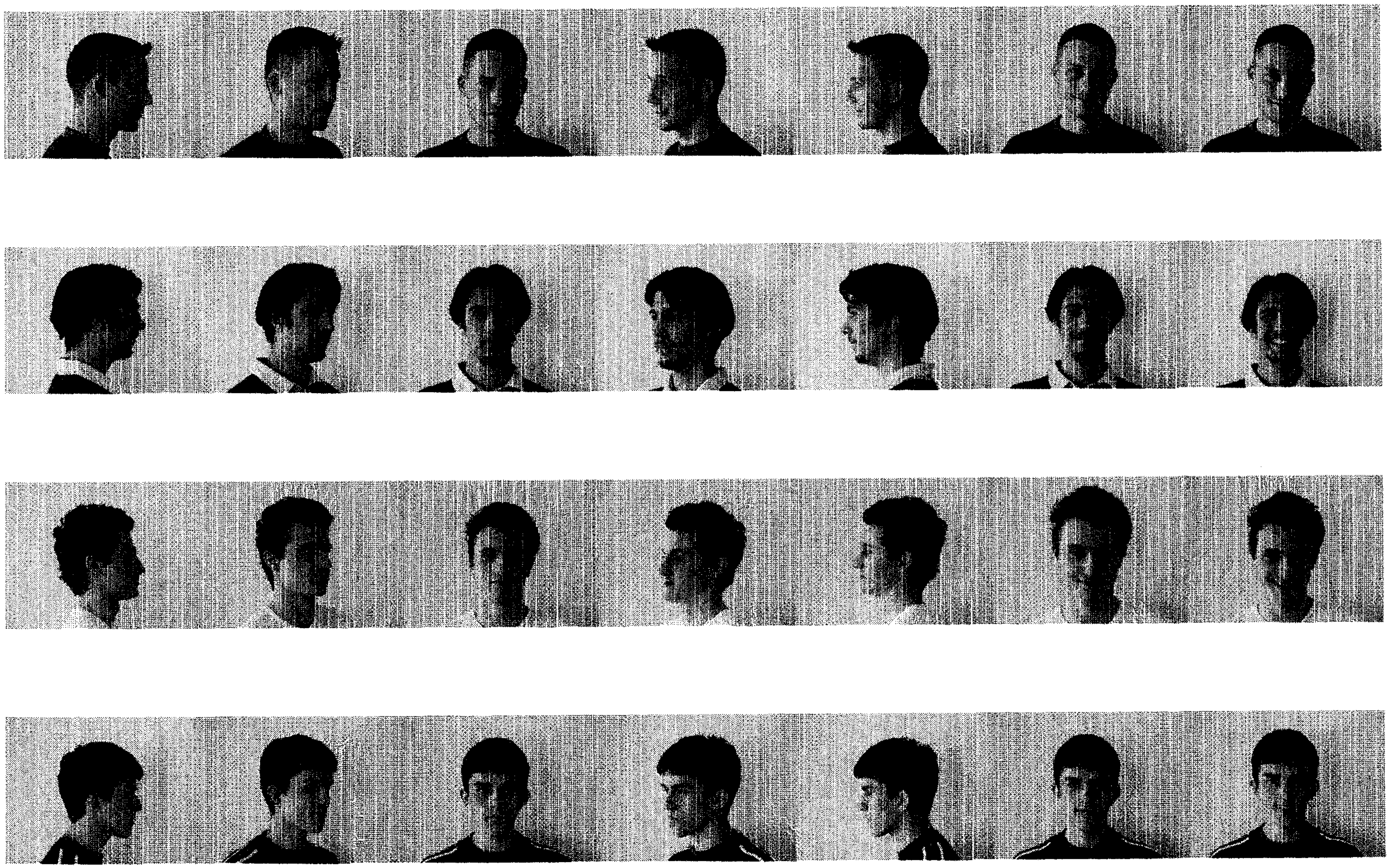


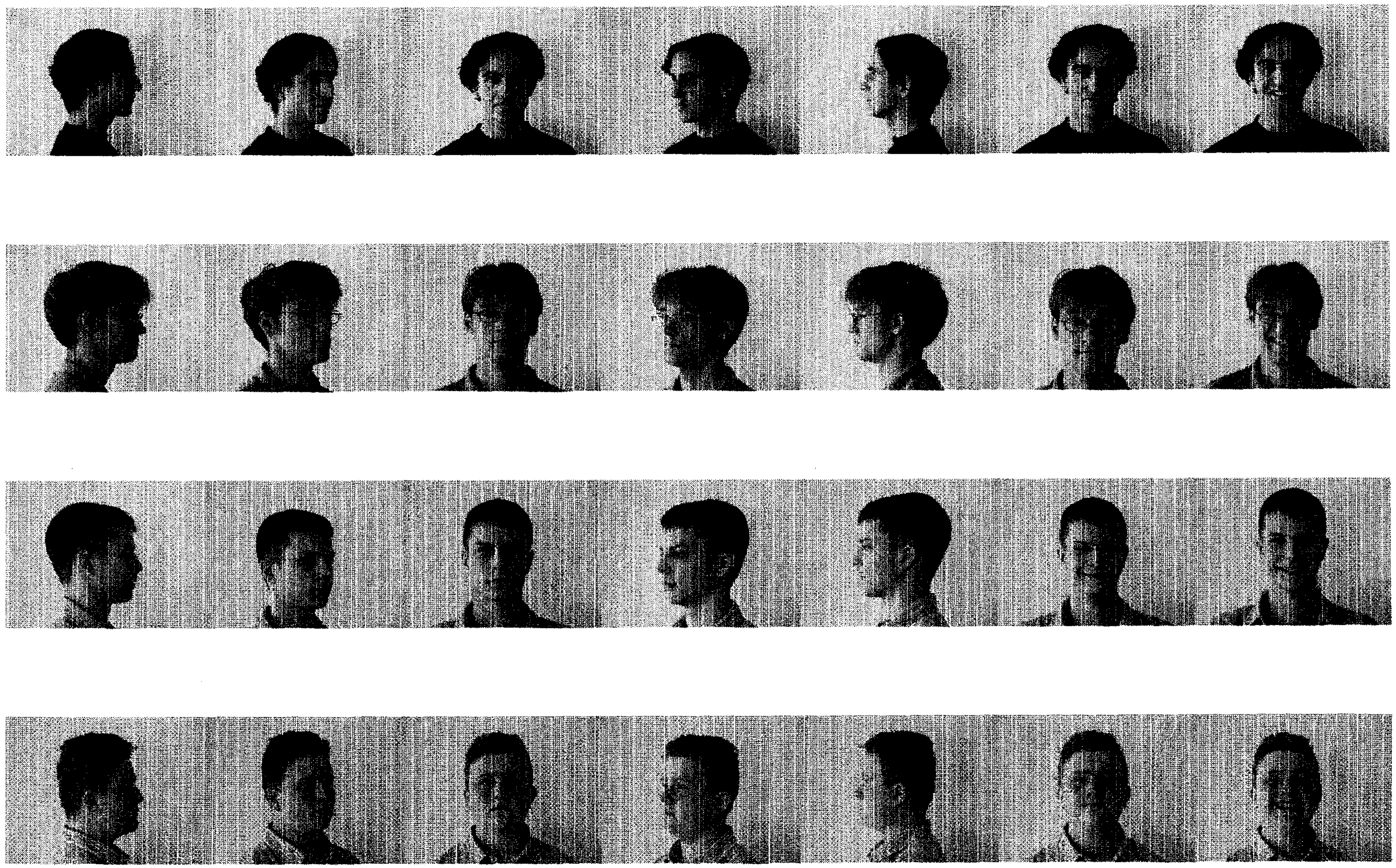




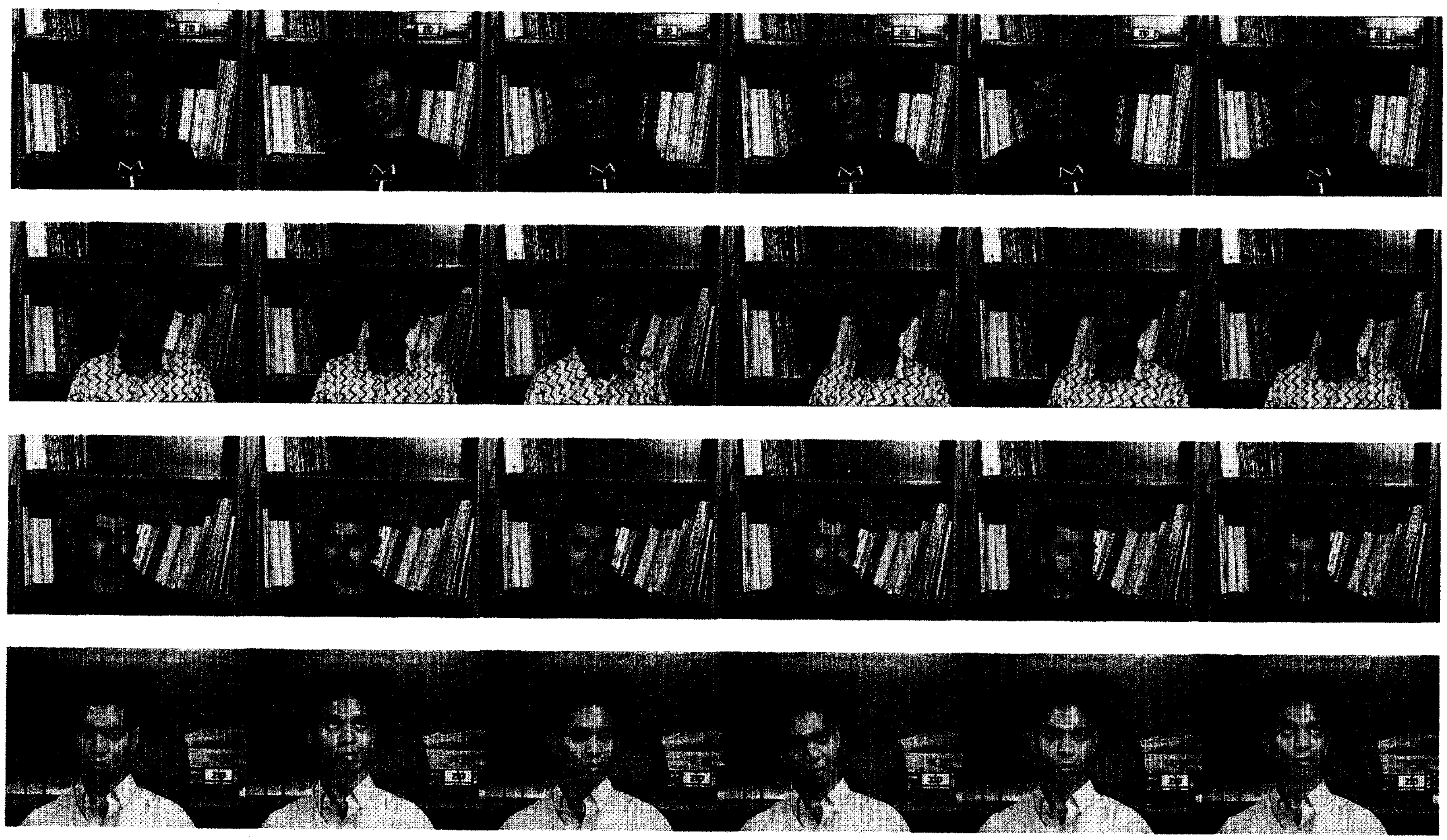
Appendix II – CVL Gray-scaled Dataset

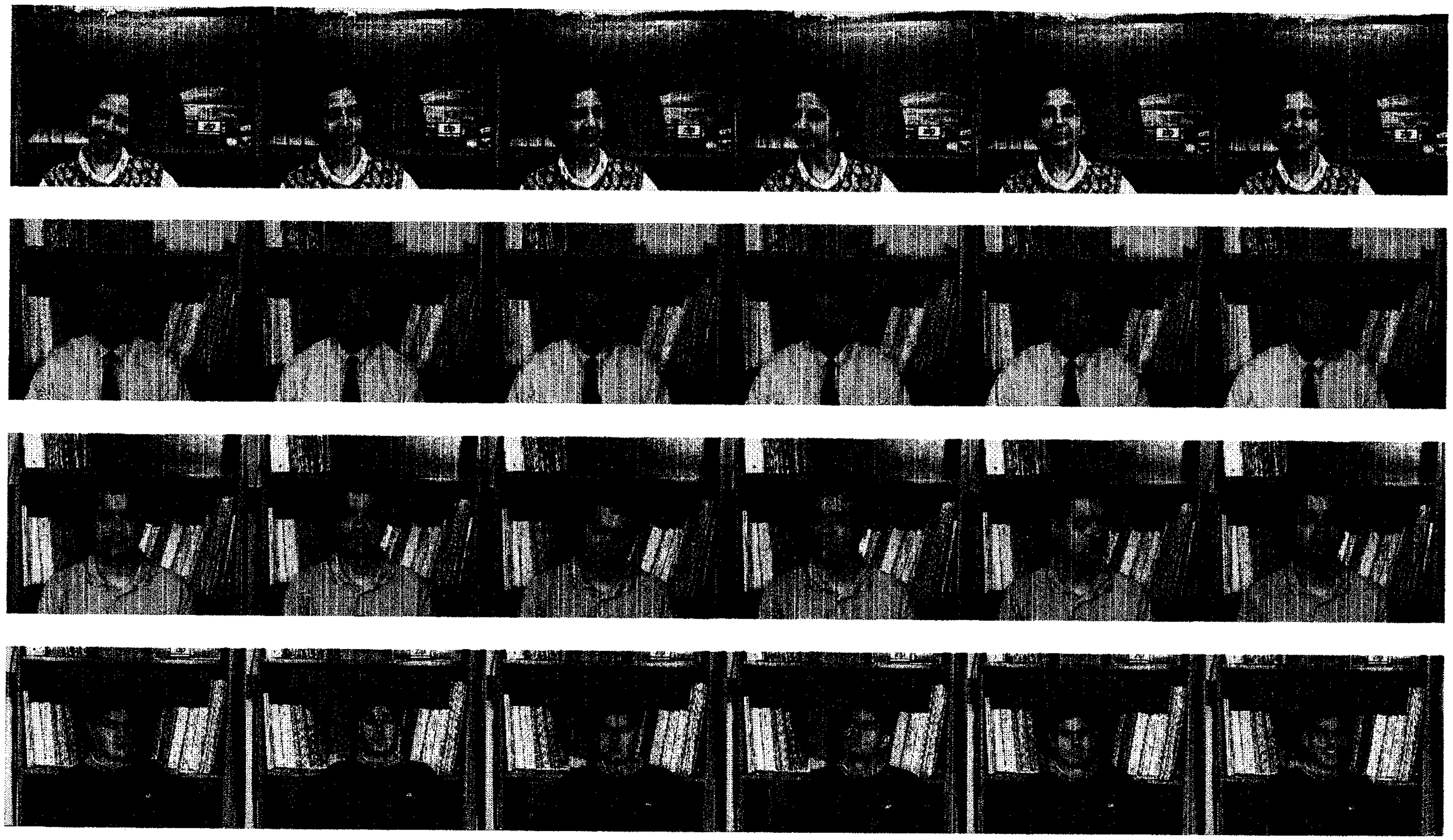






Appendix III – Georgia Tech Face Database (Color)





Appendix IV – Georgia Tech Face Database (Gray-scaled)