Fall 2015

Face Recognition: An Engineering Approach

Farshad Ghahramani
San Jose State University

FACE RECOGNITION: AN ENGINEERING APPROACH

A Thesis

Presented to

The Faculty of the Department of Computer Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Farshad Ghahramani

December 2015

The Designated Thesis Committee Approves the Thesis Titled

FACE RECOGNITION: AN ENGINEERING APPROACH

by

Farshad Ghahramani

APPROVED BY THE DEPARTMENT OF COMPUTER ENGINEERING

SAN JOSÉ STATE UNIVERSITY

December 2015

Dr. Simon Shim	Department of Computer Engineering
Dr. Yonghee Park	Department of Computer Engineering
Dr. Hyeran Jeon	Department of Computer Engineering

ABSTRACT

FACE RECOGNITION: AN ENGINEERING APPROACH

By Farshad Ghahramani

In computer vision, face recognition is the process of labeling a face as recognized or unrecognized. The process is based on a pipeline that goes through collection, detection, pre-processing, and recognition stages. The focus of this study is on the last stage of the pipeline with the assumption that images have already been collected and pre-processed. Conventional solutions to face recognition use the entire facial image as the input to their algorithms. We present a different approach where the input to the recognition algorithm is the individual segment of the face such as the left eye, the right eye, the nose, and the mouth. Two separate experiments are conducted on the AT&T database of faces [1]. In the first experiment, the entire image is used to run the Eigen-face, the Fisher-face, and the local binary pattern algorithms. For each run, accuracy and error rate of the results are tabulated and analyzed. In the second experiment, extracted facial feature segments are used as the input to the same algorithms. The output from each algorithm is subsequently labeled and placed in the appropriate feature class. Our analysis shows how the granularity of collected data for each segmented class can be leveraged to obtain an improved accuracy rate over the full face approach.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Dr. Simon Shim. I am also indebted to Dr. Younghee Park and Dr. Hyeran Jeon for advice, support, and being on the reading committee.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Face recognition has been an active area of research in the past several decades. Initially a branch of artificial intelligence to enable robots with visual perception, it is now part of a more general and larger discipline of computer vision. Computer vision applications can process images from a wide range of the electromagnetic spectrum. X-rays are used in medical technology to create images of the human body without surgery. Gamma rays and radio waves in magnetic resonance imaging (MRI) capture images of thin slices of the human body useful for diagnostic and treatment of diseases [2]. X-rays in the automotive industry are used for inspection of material that is hard to detect by the naked eye, such as casting of wheel rims for fractures, cracks, bubble-shaped voids, and defects in lack of fusion. In the food industry, X-rays and gamma rays are used for inspection, safety and quality of their products. Examples include detection of foreign objects in packaged food like fish bone in fish, contaminants in food products such as insect infestation in citrus fruits, and quality inspection for split-pits or water content distribution [3]. Figure 1 shows the electromagnetic spectrum.

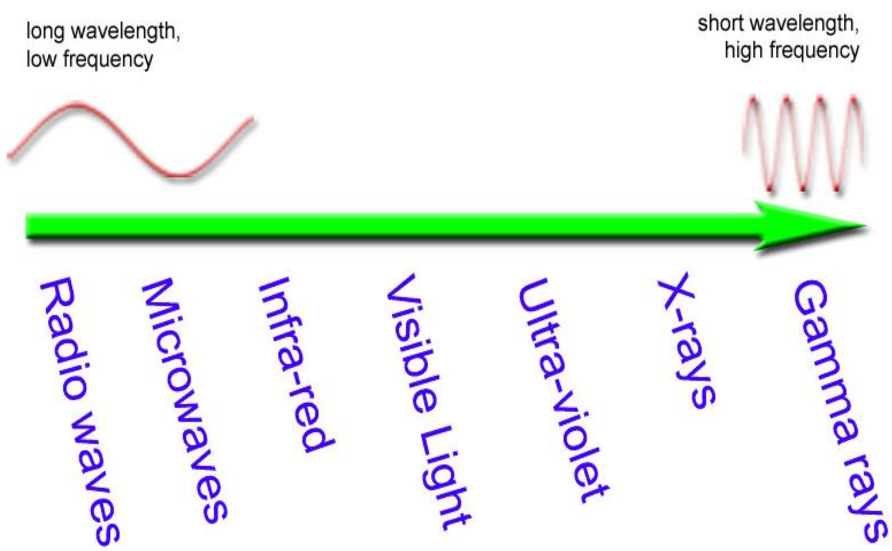
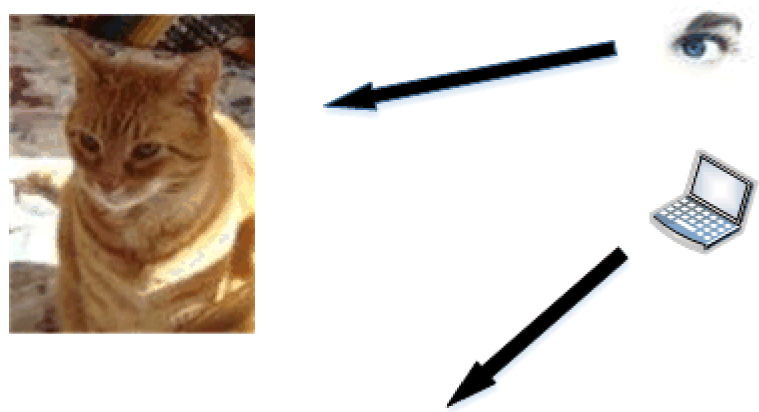


Figure 1. Electromagnetic spectrum [4].

In contrast to computer vision, face recognition applications are confined to the narrow band of visible light where surveillance and biometrics authentication can be performed. Biometrics is the term used to describe human characteristics metrics such as iris, fingerprint or hand geometry. These metrics are used for identification and access control of individuals that are under surveillance [5]. Face is becoming the preferred metric over current biometrics simply because it is a natural assertion of identity, and its non-intrusive nature provides more convenience and ease of verification. For example, in a fingerprinting system, the subject is required to interact with the system by placing a finger under a fingerprint reader, and the results must be verified by an expert. In contrast, using the subject's face as a metric requires no intervention, and the results can be verified by a non-expert.

1.1 Why computer vision is hard

All images must be first captured by a camera and then be given to a computer vision application for further processing. Compared to the human visual system, the camera is the eye, and the processing software is the brain of the application. To acquire the image, the camera uses light reflecting off an object and transmits the light intensity to its built-in sensors. The sensors then convert each of their cell intensities to a value in the range of 0-255, where a grid of numbers in this range becomes the final representation of the captured image. Note that light is a form of electromagnetic energy spanning a frequency range known as the visual spectrum. Also, sensors are unique to digital cameras as older analog cameras captured images on film. Figure 2 shows how a human sees an object like a cat and how a computer vision application sees exactly the same object.



51	52	54	54	67	69	72	76	81	80	77	72	59	50	44	33	26	31
58	51	49	50	51	65	72	81	82	85	81	84	80	73	64	56	43	33
60	58	53	49	50	57	73	80	84	88	86	90	86	84	77	68	57	44
66	66	58	52	52	54	71	75	83	91	92	92	87	88	83	75	68	54
80	76	68	60	59	59	69	72	83	90	95	96	91	91	86	80	77	63
89	84	77	73	69	68	68	74	82	87	95	98	95	96	87	85	83	69
94	93	85	85	79	78	78	76	80	87	95	100	98	97	95	91	88	77
96	92	89	89	85	86	88	84	87	90	93	99	99	97	97	93	92	82
98	91	88	85	85	87	91	94	96	95	93	96	99	99	98	96	95	90

Figure 2. Human vs. Computer vision.

The human visual system interprets the object as a cat effortlessly. It has no problem interpreting the subtle variation of translucency and correctly segmenting the object as a cat from its background. The human eye and brain are capable of extracting detailed information from the image using an existing pattern of recognition from years of experience and evolution. Furthermore, the human vision system captures objects in three dimensions with contextual properties such as depth, color, shape, and appearance. However, these properties are all lost when the camera captures an image, and its data reach a computer vision system. Given camera data as a two dimensional grid of numbers, a computer vision system has to recover the lost contextual information by inverting the camera acquisition process from unknown and insufficient information. The recovery of lost contextual properties, the visual reconstruction of an image, and its interpretation from insufficient information are the reasons that makes computer vision challenging.

1.2 Face recognition process

Face recognition is the process of labeling a face as recognized or unrecognized. The process has a life cycle based on a pipeline that goes through collection, detection, pre-processing, and a recognition stage. In the collection step, images are captured and stored for training and recognition. In the detection phase, regions of a face within an image are identified and their location is recorded. The pre-processing stage modifies the image by removing unwanted features such as shadow or excessive illumination. Recognition, the final stage of the pipeline, identifies the face as recognized or not recognized.

1.3 Face collection

Before a recognition system can identify a face, it must first be trained on a collection of images, known as the training set. The set enables comparison of its contents with a new image to determine if the difference is small enough for a positive identification. For a successful recognition, the set must be robust, meaning it must contain a variety of images such as facial images (positive samples) as well as non-facial images (negative samples) such as cars, trees, etc. Furthermore, the set must contain a variation of facial images, where the subject is looking up or down, with different facial expressions and lighting conditions. It is important to have variety in the set rather than just a large number of images with little or no variation in them.

1.4 Face Detection

Face detection is the process of locating a face in an image without identification. Although many face detection algorithms existed before 2001, a major breakthrough in face detection appeared with the Viola-Jones paper "Rapid Object Detection using a Boosted Cascade of Simple Features" [6]-[7]. Unlike previous face detection methods that relied on pixel analysis, Viola-Jones devised an algorithm called "Haar-classifier" that relied on Haar-like features. The Haar classifier is a machine learning algorithm that is trained with many positive and negative samples to detect objects in images. For the classifier to work properly, the size of the image in the training set must be the same as the size of the input image used for object detection.

1.5 Pre-processing

Face recognition algorithms are susceptible to many external effects such as head orientation, partial occlusion, facial expression, and light condition. To minimize these effects on the performance of the algorithm and to reduce error, facial images are pre-processed to make them recognition friendly. A standard pre-processing technique for reducing the effect of light condition is the histogram equalization. The image histogram is produced by a count of pixel values in the range of 0-255. If most of the high bins are to the right of the histogram, the image is bright and if most of the high bins are to the left of the histogram, the image will be dark. Equalizing a histogram distributes the bins evenly across the image, giving it a good contrast. Figure 3 (a), (b) shows histogram equalization, smoothing the intensity of light across the image. Figure 3 (a) shows dark regions are represented as high bins on the left side of the histogram. Figure 3 (b) shows how histogram equalization distributes the intensity of dark gray regions evenly across the image.

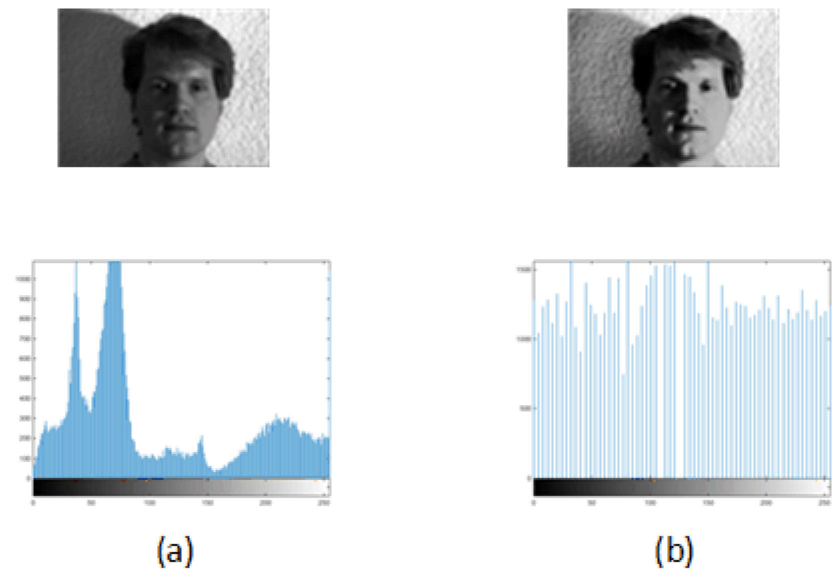


Figure 3. Histogram equalization smoothing the intensity of light across an image from the Yale database [8]. (a) without equalization. (b) after equalization.

1.6 Algorithms

The following is a list of common approaches to face recognition algorithm design.

1.6.1 Appearance based

Appearance based algorithms use image pixel data as a whole for recognition. Direct Correlation, Eigen-face and Fisher-face belong to this class of algorithms. Direct correlation uses direct comparison of image pixels of two facial images, producing a similarity score [9]-[10]. Unlike a direct correlation method that uses facial images in their original image space, Eigen-face and Fisher-face algorithms reduce the image to the most discriminating factor and make their comparison between images in a reduced dimension image space [11]-[12].

1.6.2 Active appearance

Active Appearance Model algorithms contain statistical information of an image shape and texture variation. Coots *et al.* [13] applied principal component analysis to generate statistical model that localized landmarks on the training set of images. The landmarks are used to learn displacement between a synthesized model parameter and the training images. To match an image, the current residual error of the model is measured against predicating changes to current model parameters leading to a better fit and recognition [14].

1.6.3 Support vector machines

Support Vector Machines use a training set of images to compute the optimal separating hyper plane. Guo *et al.* [15] applied this method to face recognition using a binary tree classification, where face images are iteratively classified as belonging to one of the two classes that propagates up a binary tree structure until a final classification decision can be made.

1.6.4 Bayesian model

The association of prior distribution with unknown is called Bayesian Modeling. Bayesian Model algorithms show a probabilistic measure of similarity derived from a

Bayesian Analysis of the difference between face images. Computing probability functions using the differences of image intensity in two sets of data, leads to a similarity score from Bayes rule, which can then be used for recognition classification [16].

1.6.5 Neural network

Neural networks provide information processing, that is similar to the way in which information is processed in biological systems such as the human brain. Their key strength is the ability to learn from examples, fault tolerance, and robustness. They are suited for recognition of facial images that vary a lot, and yet require little modification to the recognition algorithm. Lawrence [17] describes how to train a neural network classifier for identification and recognition of images.

1.6.6 Texture based

Texture based algorithms extract textual features from face images, by dividing a face into several regions. Local Binary Pattern (LBP) is an example of Texture based algorithms, where weighted LBP features are extracted to generate a feature vector. Two LBP feature vector are matched by applying weighted Chi-squared distance measure [18].

1.6.7 Feature based

These algorithms extract a set of geometrical features and distances from facial images and use these features as the basis of comparison between images. Local Feature Analysis is an example of feature based algorithms [19].

1.7 Data set

All our results were conducted on the AT&T data set [1]. This is a publicly available and widely used data set for face recognition research and development. The data set consists of 400 images of 40 subjects each with 10 different poses. These are single image pictures with normal lighting conditions. For some of the subjects, the images were taken at different times. The images also exhibit variation in facial expression i.e. smiling and not smiling, open or closed eyes. All the images were taken against a dark homogenous background with the subject in an upright frontal position and some degree of facial rotation up to 20 degrees. The images are all gray scale with a resolution of 92 x 112 pixels.

CHAPTER II

EIGEN-FACE

Face recognition is a measure of similarity between a new face and a set of previously observed faces in the training set. Similarity can be established by computing the difference in the distance between the images. When this difference is small, the new face is considered to be similar to one of the images in the training set, and it would be classified as recognized. If the difference is large, the new face would be considered as dissimilar to the images in the training set, and it would be classified as unrecognized. Suppose a face can be shown by only 2 pixels, and the training set contains four such images, a1, a2, b1, and b2 as shown in Figure 4.

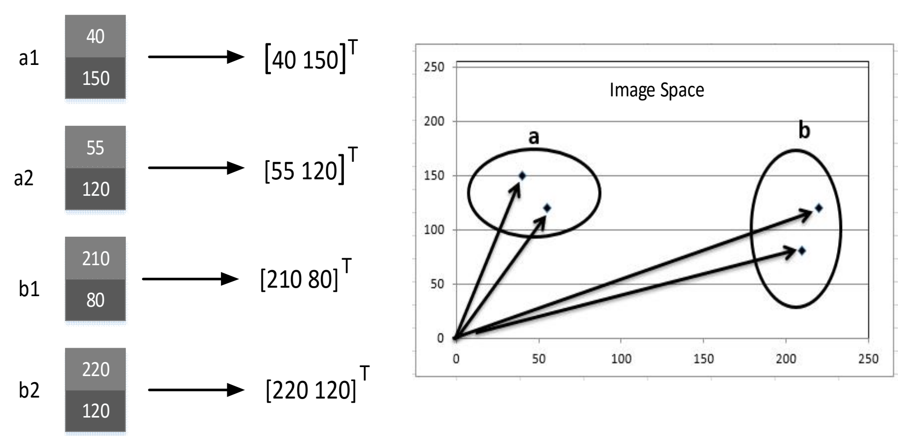


Figure 4. A two pixel image mapped to a plane.

The left side of the Figure 4 shows images a1, a2, b1, and b2 with their pixel values and their corresponding 1 x 2 transposed vector. The right side of the figure shows the transposed vector of each image mapped to a plane. From the mapped image vectors, it can be seen that images whose pixel values are close to one another are also mapped onto the plane as close to one another. A new image represented as a point in the plane can then be labeled as “a” or “b” or neither by computing its Euclidean distance from respective points in “a” or “b”. Euclidean distance between points can be computed by equation 1, where x and y are points in the plane.

$$distance = \sqrt{(delta\,x)^2 + (delta\,y)^2} \qquad \text{Equation 1}$$

Just as a two pixel image can be mapped to a two-dimensional space (the plane), larger images with more pixel values can be mapped into their respective dimension. For example, a 50 x 50 image can be mapped to 2500 dimensions whose representation is a single point in that space. Note that each pixel value represents one of the dimensions, and a vector in the space has elements in a one to one correspondence with the image pixel values. Furthermore, just like the plane, similar images in a higher dimension are closer to one another and dissimilar images will be farther apart. Computing the Euclidean distance in high dimensions involves many subtractions between the test image and a trained image. If each of these differences contributes to noise, where noise can be defined as anything that affects a positive outcome to the final recognition, then the total number of noise will be very high. This is because summing all squared differences will contribute a lot to the noise that would be high compared to the amount of useful information. Since computing the difference between pixel values in a higher dimension is not practical due to image noise, a dimensionality reduction of the original image space has to be performed. Similarity between images will then be based on how the image points are spread in this reduced dimensional space [20].

2.1 Dimension reduction

Very often, the information of interest can be found in a lower dimension than the original image space. The dimensionality reduction approach brings out useful information that can be revealed in lower dimensions which is demonstrated for the best line fit in Figure 5.

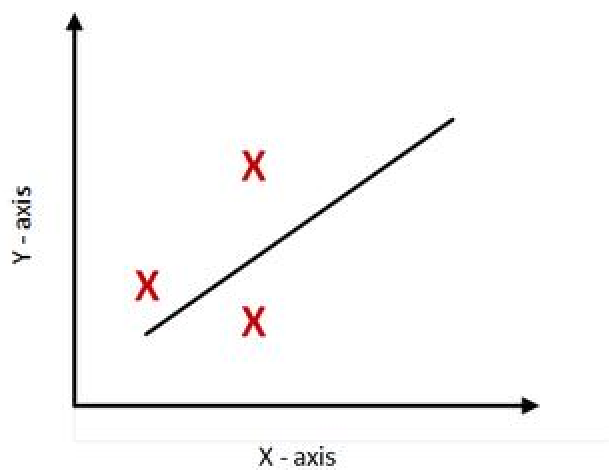


Figure 5. Best line fit.

Among all the lines that pass through the points in Figure 5, there is only one line for which the distance between the points and the line is a minimum. This line, the best fit line has several properties. First, it represents a relation between the three points. For example, the three points could represent three homes where a utility company plans to lay down lines with a minimum cost. Second, the line is a one-dimensional object, representing the transformation of points from two-dimensional space to something meaningful in a one-dimension space. The idea of transformation of points from a higher dimension space to a lower dimension space is also used in face recognition algorithms. For example, the Eigen-face recognition algorithm uses principal components analysis to reduce an image space down to the most variant feature by projecting it to a lower dimension subspace, where face recognition is performed.

2.2 Subspace

The "best line fit" as described in section 2.1, is a one-dimensional object and since it is found inside a plane, a two-dimensional object, it would be a subspace of a two dimension space. Furthermore, because the line is fitted through the points, its slope indicates the direction in which the points are spread out the most. Placing a coordinate system with the origin anywhere on the line captures the variation of points within the context of a new reference frame. The line, given by the equation of $y = mx + b$, becomes a subspace of two-dimension space defined by the x-y coordinate system. The new reference frame emphasizes the most interesting aspect of data, which is the direction the points are separated from one another [21]. The direction of maximum separation is called the first principal component of the data set. The next largest separation is a line perpendicular to the first, and it is known as the second principal component. Figure 6 shows a new reference frame drawn upon the distribution of points within the x-y coordinate system. From the figure, it can be seen that the distribution of points is not completely random, and there is a linear relation between x and y values. When x values are large, y values are large, and when x values are small, y values are also small.

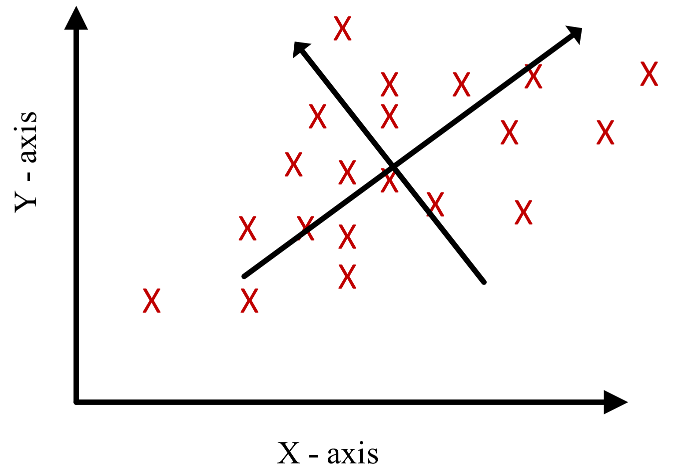


Figure 6. Reference frame drawn upon distribution of data.

Figure 7 shows the new reference frame displayed on its own coordinate system, with the first principal component as the horizontal axis and the second principal component as the vertical axis.

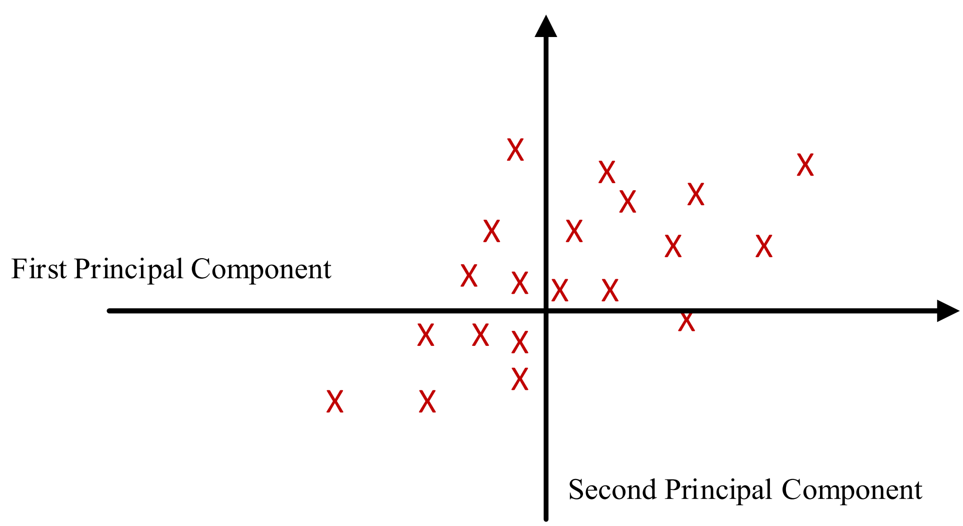


Figure 7. Principal Component Coordinates.

From the figure, it can be seen that the first component varies over a wide range of points, while the second component varies over a more restricted range. Note that points that are not part of variation along the first and the second principal component will be of no interest and are not represented in the principal component coordinate system. Therefore, sub-spacing an image from a high to a low dimension removes unwanted data points and only keeps those that are influential to the outcome of the final result. For the purpose of comparison, variations of data points that describe facial features are important, but other changes in the data, such as light and illumination, are non-factor in comparison and are irrelevant to the recognition outcome. In this case, the variation in light are the points that will be removed by sub-spacing and dimension reduction, whereas facial features are the points that will be kept along the principal component coordinate system.

2.3 Principal component analysis (PCA)

PCA is a statistical technique for finding patterns in high dimension data such that their similarities and differences are highlighted. It transforms data from their original coordinate system to a new coordinate system, where major distribution of points is along the first principal component. The next largest variation of data is mapped along the second principal component perpendicular to the first principal component axis [22]. PCA is an effective technique for finding patterns of similarity and dissimilarity in face recognition, mainly because finding patterns in high dimension data is difficult, and images are represented by points in high dimension space.

2.4 Face detection

Before an image can become part of the training data set, it must first be detected. We used Viola-Jones Haar classifier for face detection, which was trained from face and non-face objects, with the information stored in XML file. To apply the classifier correctly, several factors must be considered. First, it is important to convert color images to gray, since face detection only works on gray scale images. Second, the speed of face detection depends on the size of the input image. Face detection can become very slow for large images, but fast for small images. Third, a low illumination of light can affect the result of a face detection algorithm. Our data set contained images that were all gray level with a reasonable small size of 92 x112 pixels and with a uniform intensity of light. As a result, this data set did not require any pre-processing.

2.5 Training

Once the face detection step is complete, the detected face can be added to the training set. Our training set contained multiple images of each person, providing examples of how a person's face may change from one image to another. The changes were in frontal face orientation, illumination, and facial expression. The training set contained 7 images from each 40 subjects, with 280 images in total. To use a face recognition algorithm correctly, several factors must be considered. First, the size of the test image and those of the trained images must be the same. If a test image is reasonably larger than the training set images, it can be resized to a smaller size, while keeping the aspect ratio of the larger image the same as the smaller image. Aspect is the ratio of the height to the width of an image. Without aspect ratio adjustment, the resized image may be elongated in the vertical or horizontal direction or both causing an adverse effect on recognition algorithm accuracy. If the test image is reasonably smaller than the images in the training set, it can be enlarged to the same size as the images in the training set. Enlarging the image may cause distortion, blur, or pixilation, all adversely affecting the recognition algorithm accuracy [23]. Second, the alignment of the test face should be as close as possible to the alignment of the faces in the training set. If the training set contains faces that look straight into the camera, such as ID photos, and the test image is of a person looking up or down, left or right, then the recognition algorithm may not be able to accurately recognize the face, as it may be comparing part of an eye with a nose. Third, facial expression in the training set should be as varied as possible. If the training set contains only faces with closed eyes in a frowning facial posture, and the test image is of a face with open eyes, then the face recognition algorithm may not be able to recognize the face. Fourth, the effect of light should also be considered. A training set with images where light illumination is high on one side will create a shadow on the other side of the face. A test image whose light illumination on the side of the face is opposite to that of the training set will have a shadow where training images on the same side of the face are bright and lack shadow. In this case, the recognition algorithm will be comparing dark regions against light regions and may fail to recognize the image. For this reason, it is important to have a uniform illumination across all images in the training set and the test images.

2.6 Algorithm

Eigen-face is one of the most well-known face recognition algorithms. It has been described by Turk and Pentland [11] in their 1991 published paper "Face Recognition Using Eigen faces." The principle of their algorithm is based on PCA, where gray level images are reduced down to the most variant feature by projecting them to a lower dimension subspace. Recognition between images is performed using distance based matching method. If the distance between the new face and the faces in the training set is small and above a threshold, the new face will be classified as known. Otherwise, the new face would be classified as unknown. The following is a list of the steps for Eigen-face algorithm:

1. Find the mean across all images. Mean μ is given by equation 2 :

$$\mu = \frac{1}{M} \sum_{k=1}^{M} \mathrm{Xi} \qquad \text{Equation 2}$$

where Xi is one of the vectors in the training set. Recall that images are represented by vectors whose elements are the pixel values of the image. The purpose of this step is to reduce noise, where noise can be defined as any feature that does not contribute to the overall recognition accuracy.

2. Compute the covariance matrix S from equation 3:

$$S = \frac{1}{M} \sum_{k=1}^{M} (\mathrm{Xi} - \mu)(Xi - \mu)^{Ti} \qquad \text{Equation 3}$$

where T_i is the transposed vector. The significance of covariance matrix is not so much as the value it contains, but as the sign of those values, since the diagonal of the covariance matrix will show the direction in which data is changing. A positive value of covariance shows that dimensions increase or decrease together. A negative value indicates when one dimension increases, the other decreases, and a value of zero shows that the dimensions are independent of one another.

3. Compute the Eigen values λ_i and Eigen vectors v_i of covariance matrix S.

$$S V_i = \lambda_i v_i \qquad \text{for } i = 1, 2 \ldots n \qquad \text{Equation 4}$$

4. Order Eigen vectors by their Eigen values. Eigen vectors with small Eigen value are less significant than those with higher Eigen value and can be simply ignored. Eigen vectors with higher Eigen values are the principal component of data.
5. Project all training images into PCA subspace.
6. Project the query image into PCA subspace.
7. Compute the smallest distance between the projected query image and the training image.

2.8 Results

Table 1. Eigen-face test run summary.

Number of Trained Images	280
Number of Test Images	120
Number of correct recognition	114
Number of failed recognition	6
Accuracy	95%

Table 2. Eigen-face failed recognition on images from the AT&T database [1].

Actual Subject	Recognized Subject
Subject 1 pose 8 	Subject 24 pose 4 
Subject 5 pose 10 	Subject 40 pose 5 
Subject 10 pose 10 	Subject 38 pose 4 
Subject 16 pose 8 	Subject 30 pose 6 
Subject 28 pose 8 	Subject 37 pose 7 
Subject 40 pose 10 	Subject 5 pose 1 

CHAPTER III

FISHER-FACE

Fisher linear discriminant analysis was first developed by Robert Fisher [24] in 1936 for the classification of flowers. It is a classical technique in pattern recognition that performs a class dimensionality reduction. The principal idea is that similar classes are clustered together, while different classes will be scattered as far away as each other. Belhumeur *et al.* [10] successfully applied Fisher linear discriminant analysis to face recognition, using a linear projection onto a low dimension subspace. In contrast to the Eigen-face, which maximizes the total variance within classes across all faces, the Fisher-face approach confines the variance within classes to the classes themselves. This results in minimizing the spread of variance to other classes. For example, by using multiple facial images of the same person, where one of the face images is with an open mouth, the open mouth discriminating factor would be confined to the images of this person only.

3.1 Difference between Eigen-face and Fisher-face algorithms

Both the Eigen-face and the Fisher-face algorithms work on the same principle of reducing the image dimension down to the most discriminating factor, where further analysis can be performed. The Fisher-face algorithm uses inner class information for face classification. It can use multiple faces of a person to establish in-class variation in order to maximize class separation. In contrast, the Eigen-face algorithm uses one image per person, thus applying the variation in one image to the entire recognition process.

The unwanted consequence of spreading the total variance in the Eigen-face algorithm leads to retaining undesirable effects such as illumination or facial expressions [12].

3.2 Linear discriminate analysis (LDA)

LDA is a statistical technique to classify objects into mutually exclusive groups based on a set of unique features. The features are the observed faces and the groups can be classified as recognized and unrecognized. Discriminant refers to those features that may describe the group, such as recognized and unrecognized. Linear means that groups are separable by a linear combination of features that describes the objects. If there are only two features, then the separation between the object groups becomes a line. For three features, the separator is a plane and for more than three features, the separator would be a hyper plane. Similar to PCA, LDA is used as a dimensionality reduction technique to project a dataset onto a lower-dimensional space. However, in addition to finding a new reference frame that maximizes the variance of data, LDA seeks to find a coordinate axis that maximizes the separation between multiple classes as shown in Figure 8.

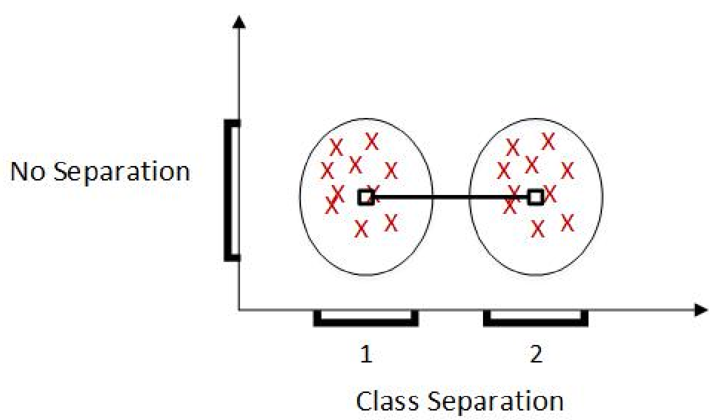


Figure 8. LDA class separation.

From Figure 8, it can be seen that without LDA, classes are mapped to the vertical axis where the separation between classes will be lost. However, by using LDA, classes are mapped to the horizontal axis where class separation is preserved. Note that each class contains in-class variation. For example, the first data set contains faces with an open mouth, while the second represents those with a closed mouth.

3.3 Algorithm

Similar to the Eigen-face algorithm, the training images are first projected onto a subspace. Then, a test image is projected onto the same subspace for a similarity measure. However, subspace is measured differently as it is outlined in the following steps:

1. Calculate within class variation. This is a measure of the amount of variation between items in the same class.

2. Calculate between class variations.

3. Compute Eigen vectors and Eigen values of within class and between class variations.

4. Sort the Eigen vectors by their associated Eigen values from high to low, and keep the highest value Eigen vectors. These Eigen vectors are the Fisher basis vector.

5. Calculate the dot product of images with the fisher basis vector. This calculation projects images onto a subspace.

6. Once trained images are projected onto a subspace, they must be collected and categorized as how close they are to one another. Computing the distance between images can establish their similarity. For example, images with the smallest distance between them can be considered close and similar, whereas images whose distance are farther apart would be considered dissimilar.

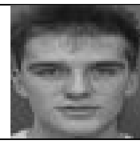
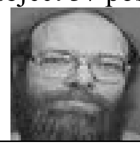
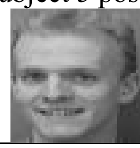
3.4 Results

Table 3. Fisher-face test run summary.

Number of Trained Images	280
Number of Test Images	120
Number of correct recognition	114
Number of failed recognition	6
Accuracy	95%

Table 4. Fisher-face failed recognition on images from the AT&T database [1].

Actual Subject	Recognized Subject
Subject 1 pose 8 	Subject 24 pose 4 
Subject 5 pose 10 	Subject 40 pose 5 
Subject 10 pose 10 	Subject 38 pose 4 
Subject 16 pose 8	Subject 30 pose 6

	
Subject 28 pose 8 	Subject 37 pose 7 
Subject 40 pose 10 	Subject 5 pose 1 

The results of Fisher-face are identical to the results of the Eigen-face recognition algorithm in section 2.8. This can be attributed to Martinez and Kak [25], who uncovered the accuracy of LDA and PCA based algorithms are dependent on adequate training data set. They suggest if the training set is not representative of the image space regions occupied by individual subjects, then the overall image variance can be a better discriminator than optimizing the ratio of between class and within class variance. Note that our training set contained 280 images, whereas the image space has a dimension of 92 x 112 or 1034.

CHAPTER IV

LOCAL BINARY PATTERN

Local binary pattern (LBP) is a texture based algorithm. Texture is an important characteristic of images. It provides information about physical properties of objects like smoothness, roughness, or difference in surface reflectance such as color [26]. Using texture to capture physical properties of objects was first proposed by Wang *et al.* [27], who encoded information in an image by mapping the local neighborhood surrounding pixel values. Continuing with the idea of encoding information in local neighborhoods, Ojala *et al.* [28] developed the LBP operator for encoding texture and shape description for digital images. The LBP operator processes an image as a composition of small patterns whose histogram reveals information about the distribution of edges and other local features. The term "operator" refers to a mapping function that can transform an image from one form to another. In 2004, Ahonen *et al.* [29] successfully applied the LBP operator to face recognition by dividing an image into regions from which LBP features were extracted and concatenated into enhanced feature vectors. The term "feature extraction" refers to the process of transforming an image into a set of features significant of the relevant properties of the original image and capable of summarizing them in a compact form.

4.1 How it works

LBP tests the relation between a pixel and its neighbors, encoding this relation into a binary word as shown in Figure 9.

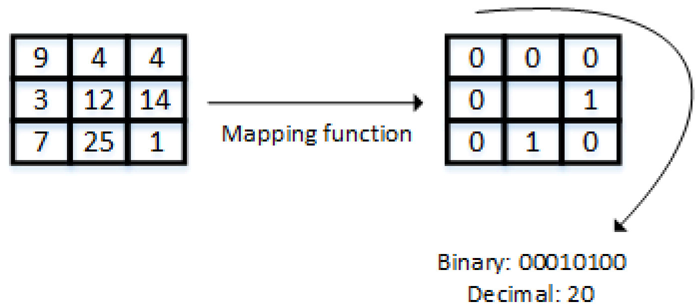


Figure 9. LBP operator on center pixel as a threshold.

For every pixel in the image, the relation between a center pixel and its neighborhood is encoded as an LBP value. These values represent the new and the transformed image used to compute the distribution of local LBP in histograms as a feature that characterizes the global texture of the image. Note that in a 3 x 3 neighborhood, there are $2^8 = 256$ different labels that can be used as a texture descriptor and as distinct bins in a histogram. Similarity of regions between images can be obtained by histogram comparison from the Chi-squared, the log-likelihood ratio, the histogram intersection, or the Jenison Shannon divergence test. The final similarity for the whole image would be the sum of all regional similarities [29].

4.2 Algorithm

The LBP algorithm can be outlined as regionalizing an image and using its statistical distribution to provide local texture representation. More specifically, the algorithm first extracts and trains visual features, and then summarizes their distribution. The list of steps in the algorithm are as follows:

1. Divide the image into non-overlapping local binary map rectangular regions, e.g. 10 (2 x5) or 40 (3 x 3) or 16 (4 x 4) etc. Figure 10 shows how an image can be divided into 3 x 3 neighborhood regions.

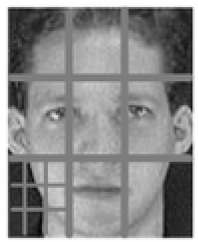


Figure 10. LBP for a 3 x 3 neighborhood regions.

2. For all the neighborhoods in the image, compute the LBP value based on a suitable threshold. Thresholds are usually set manually to obtain the best performance for a specific problem, but they can be set automatically by exploiting the local statistics as the mean and standard deviation for each neighborhood [30].

3. Compute histograms of LBP values for all the rectangular regions as shown in figure 11.

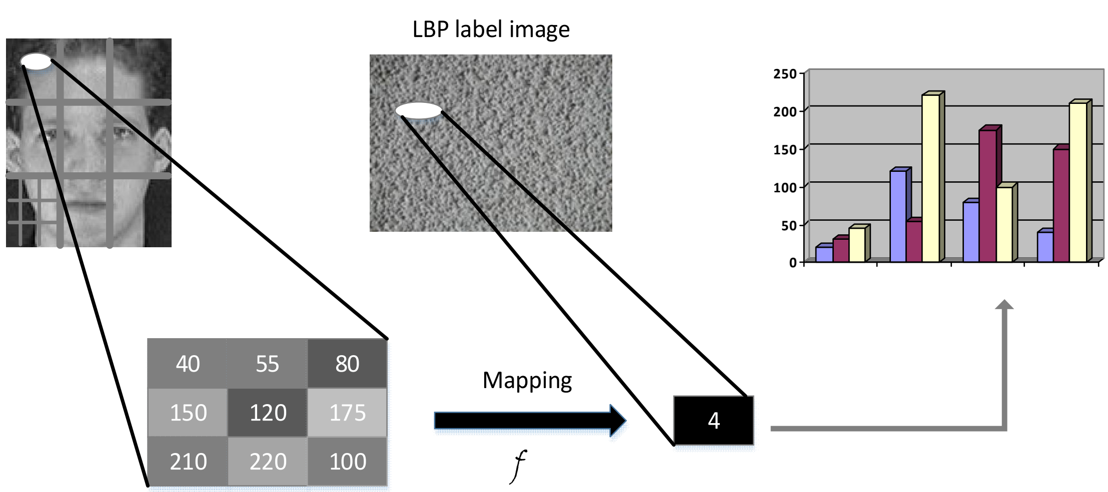


Figure 11. LBP local binary map.

4. Perform recognition of the test image using the nearest neighbor classifier where the similarity measure can be any of the following: histogram intersection, log-likelihood statistics, or Chi-squared [31]. For example, Chi-squared can be computed by:

$$\chi^2\,(S,M) = \sum_{b=1}^{B} \frac{(S_b - M_b)^2}{(S_b + M_b)} \qquad \text{Equation 5}$$

where S and M denote sample and model distribution. B is the number of bins in the distribution, S_b, and M_b correspond to the probability of bin b in the sample and model distribution. A Chi- squared value of 0 indicates a perfect match with numbers closer to 0 indicating a better match than larger values.

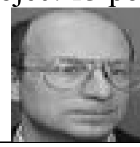
4.3 Results

Table 5. LBP test run summary.

Number of Trained Images	280
Number of Test Images	120
Number of correct recognition	110
Number of failed recognition	10
Accuracy	91%

Table 6. LBP failed recognition on images from the AT&T database [1].

Actual Subject	Recognized Subject
Subject 1 pose 8 	Subject 13 pose 6 

Subject 1 pose 9 	Subject 4 pose 5 
Subject 1 pose 10 	Subject 13 pose 6 
Subject 5 pose 9 	Subject 21 pose 7 
Subject 10 pose 9 	Subject 9 pose 7 
Subject 10 pose 10 	Subject 4 pose 4 
Subject 16 pose 8 	Subject 27 pose 1 
Subject 28 pose 8 	Subject 18 pose 1 
Subject 29 pose 9 	Subject 23 pose 6 
Subject 39 pose 10 	Subject 12 pose 3 

CHAPTER V

FACIAL FEATURE SEGMENTATION

Facial features can be identified by using geometry and their relative position to one another. For example, the thickness of the eyebrow and its relative position to the eye can be measured to identify its location. Brunelli and Poggio [32] describe a recognition system based on geometrical features to distinguish between faces. They apply a template matching technique to locate a feature, utilizing the knowledge of the average face structure to refine the search for the remaining features. Once all the features are detected, a dimensional vector is created to represent the face. Recognition is then carried out by means of the nearest neighbor classifier.

A different approach is presented in this study. Unlike Brunelli and Poggio, there will be no attempt to make any measurement of facial features. Instead, the facial features are segmented and extracted out of the image and placed in their own data set. Although pre-processing techniques as described in section 1.8 can improve the accuracy of the results, a conscious decision has been made to factor out this step and its influence on the outcome of the face recognition approach.

5.1 Features and data set

The training data set contained 7 images for each 40 subjects in the AT&T data set without any overlap. Each subject was further divided into the left and the right eye, the nose, the mouth, and the both-eye data sets. A sample of subject 1's features in the training data set is shown in table 7.

Table 7. Feature segmentation, Subject 1.

Segment	Segmented image	Size in pixels	Training set size
left eye		27 x 27	280
right eye		27 x 27	280
Nose		32 x 32	280
Mouth		42 x 25	280
Both-eye		65 x 15	280

Test images were chosen from the remaining 120 subjects in the AT&T data set. Each of the 40 test subjects were in 3 different poses making up the total 120 images. The feature segmentation algorithm for the test subjects was set to be the same as those for the training set. For example, the procedure to segment a subject's left eye for training was the same as that for testing. Since there is more than one classification of data, i.e. "left eye," "right eye," facial feature segmentation provides a better granularity than full face recognition. In the following sections, a discussion of data granularity shows how it can be leveraged to improve recognition accuracy among the Eigen-face, Fisher-face, and LBP algorithms.

5.2 Feature distribution

The distribution of recognized features varied for each applied recognition algorithm, but it stayed mostly within 1-3 recognition levels as shown in Figures 12-14, 18-20, 23-25, and 30-35. In the figures, the horizontal axis is set to be a common scale for subjects 1 through 40, with their vertical axis displaying how a particular category of data is distributed.

5.3 Left eye distribution

The left eye distribution is shown in Figures 12, 13, and 14. The Eigen-face and Fisher-face are mostly within 2-3 recognition levels, and LBP is within a 1-2 range.

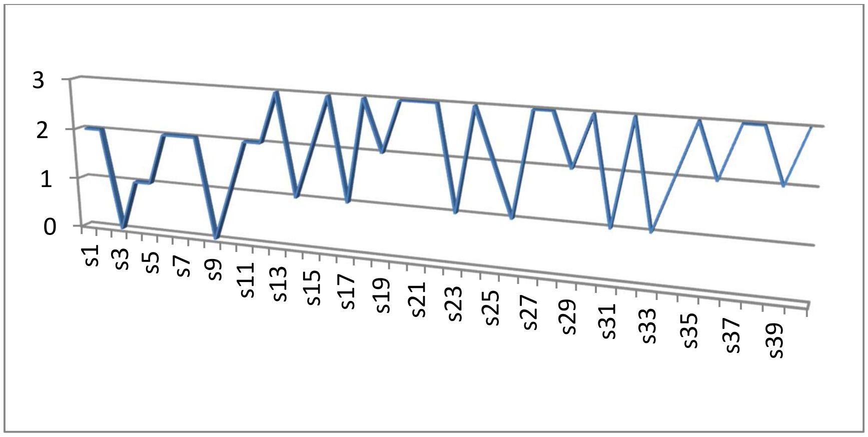


Figure 12. Left eye recognition distribution, Eigen-face algorithm.

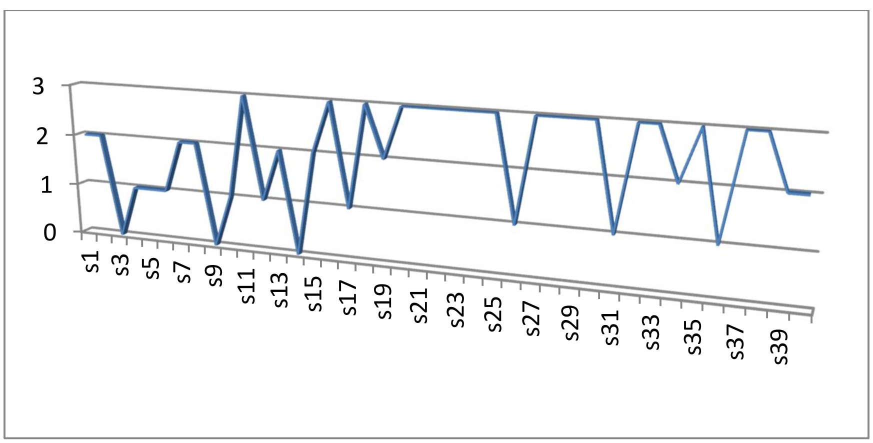


Figure 13. Left eye recognition distribution, Fisher-face algorithm.

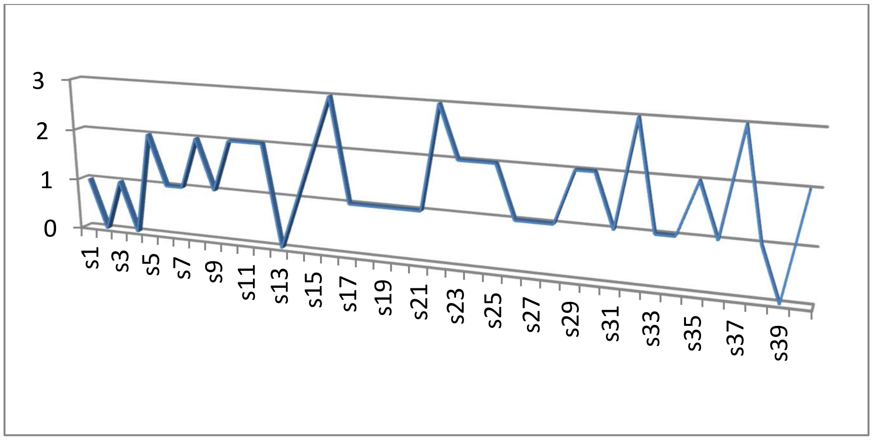


Figure 14. Left eye recognition distribution, LBP algorithm.

The left eye recognition failure among the 3 algorithms is summarized in table 8.

Table 8. Failed left eye recognition.

Algorithm	Subject left eye
Eigen-face	3,9
Fisher-face	3,9,14
LBP	2,4,13,39

As discussed in chapter 2 and 3, the Eigen-face and the Fisher-face algorithms are very similar except in their approach to image analysis in reduced dimensional space. Both algorithms failed on Subject 3 and Subject 9, but the Fisher-face additionally failed on Subject 14. Test subjects 3, 9, and 14 are shown in Figure 15, 16, and 17.

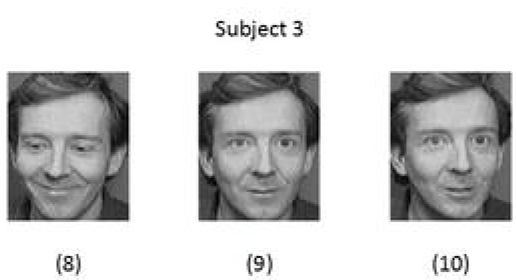


Figure 15. Subject 3 in left eye recognition test. Subject 3 image is from the AT&T database [1].

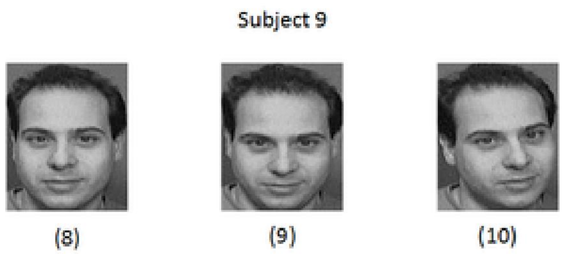


Figure 16. Subject 9 in left eye recognition test. Subject 9 image is from the AT&T database [1].

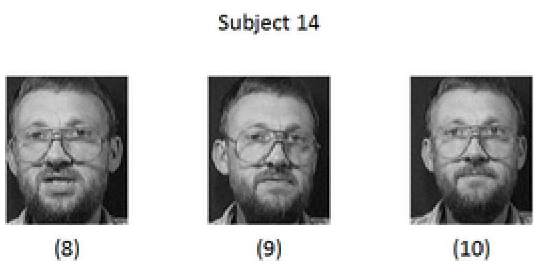


Figure 17. Subject 14 in left eye recognition test. Subject 14 image is from the AT&T database [1].

Note that in the above images, Subject 14 (8) (9) (10) is wearing glasses, Subject 3 (8) is looking down, and Subject 9's (10) eyes are not aligned with the camera. From Table 8, it can be seen that Eigen-face algorithm has outperformed both the Fisher-face and LBP algorithms for the left eye recognition.

5.4 Right eye distribution

The right eye distribution in Figures 18, 19, and 20 shows Eigen-face and Fisher-face are mostly within 2-3 recognition levels while LBP is within a 1-2 recognition range.

This is due to the fact that LBP needs more regions for its grid computation than our 27 x 27 pixel size for the right eye segmentation.

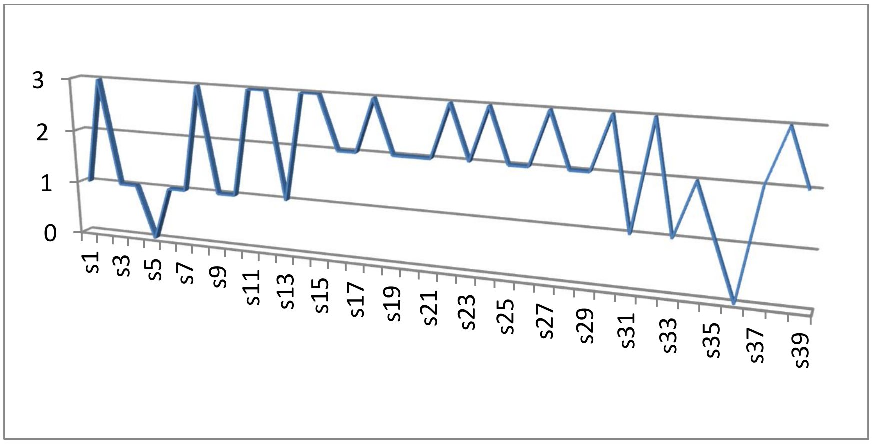


Figure 18. Right eye recognition distribution, Eigen-face algorithm.

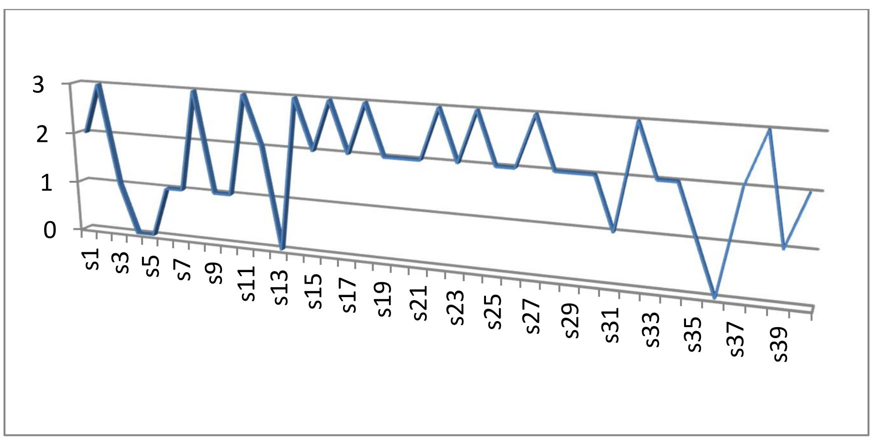


Figure 19. Right eye recognition distribution, Fisher-face algorithm.

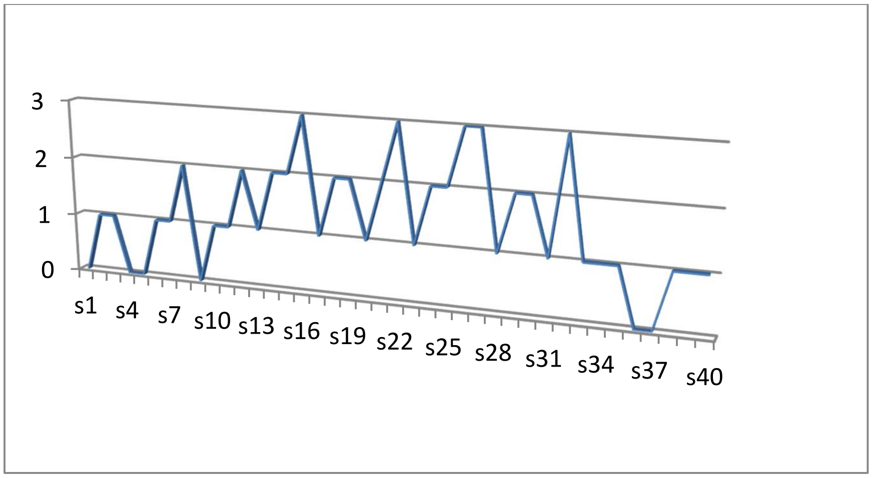


Figure 20. Right eye recognition distribution, LBP algorithm.

The right eye recognition failure among the 3 algorithms is summarized in Table 9.

Table 9. Failed right eye recognition.

Algorithm	Subject right eye
Eigen-face	5,36
Fisher-face	4,5,13,36
LBP	1,4,5,9,36,37

From Table 9, it can be seen that Eigen-face has outperformed both the Fisher-face and the LBP algorithm for the right eye recognition. Note that all the three algorithms failed on Subjects 5 and 36 as shown in Figure 21 and 22 respectively.

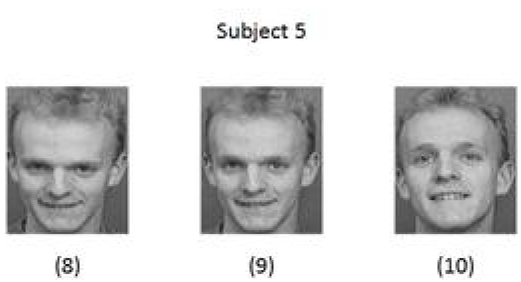


Figure 21. Subject 5 in right eye recognition test. Subject 5 image is from the AT&T database [1].

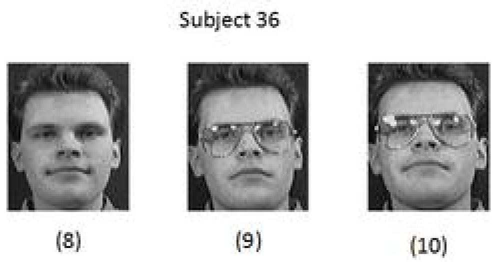


Figure 22. Subject 36 in right eye recognition test. Subject 36 image is from the AT&T database [1].

The training set for Subject 36 contained 7 images all without glasses, but two of the test images in Figure 36 are wearing glasses. As mentioned in chapter 1, partial occlusion of the face can affect the accuracy of recognition algorithms and this could be the reason for the right eye failure in this case. A remedy to this problem is to run a classifier designed specifically for detection and recognition of eye glasses. Although such classifier was available, its application was bypassed to keep the results as pure as possible. Note that none of the algorithms had any problem recognizing the left eye of subject 36, but they all failed on the right eye. In a full face recognition scenario as presented in chapters 2, 3, and 4, an unrecognized image has no recourse for further processing. However, in the segmentation approach, a full face is made up of several features, each providing more opportunity for a positive recognition outcome.

5.5 Both-Eye distribution

The both-eye distribution in Figures 23, 24, and 25 shows Fisher-face algorithm displaying better results than both Eigen-face and LBP.

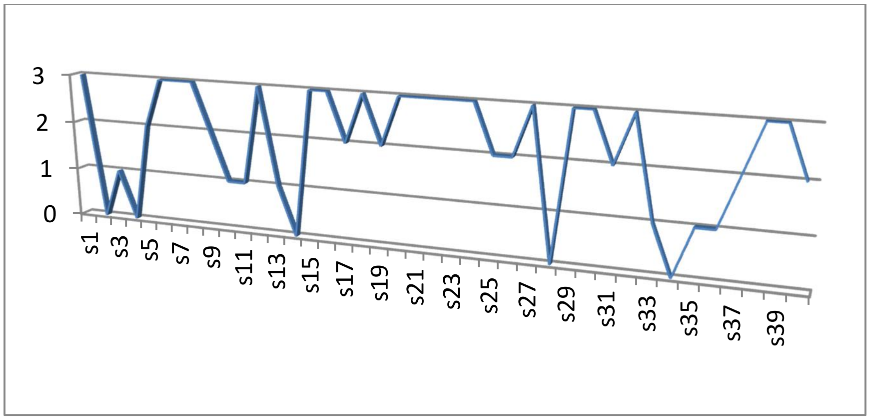


Figure 23. Both-eye recognition distribution, Eigen-face algorithm.

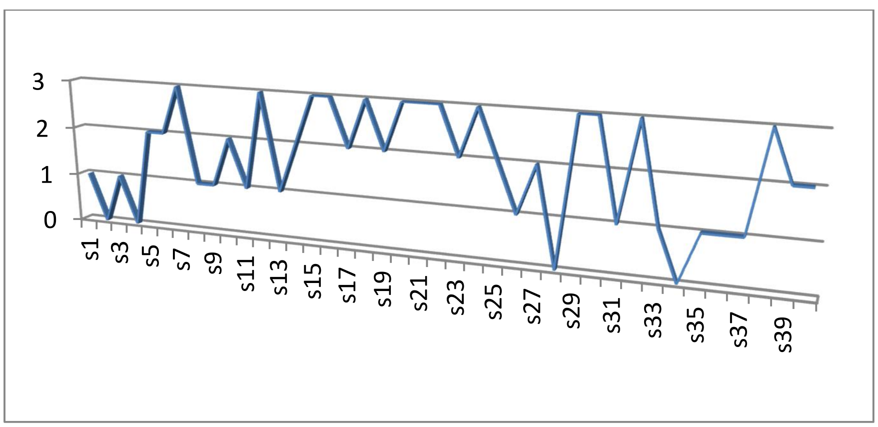


Figure 24. Both-eye recognition distribution, Fisher-face algorithm.

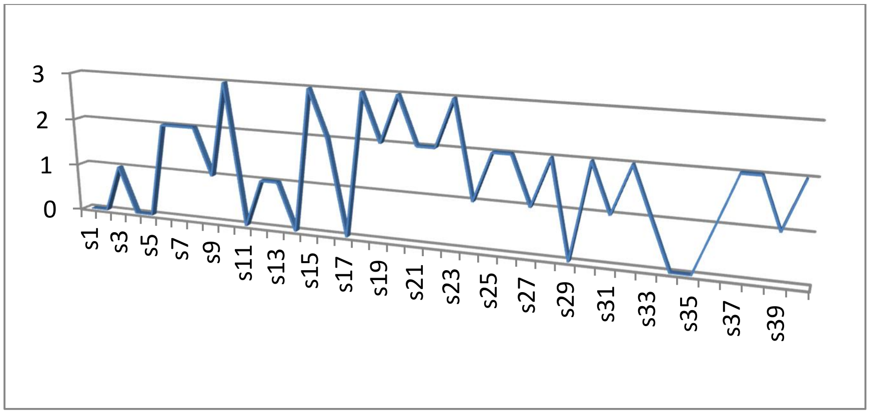


Figure 25. Both-eye recognition distribution, LBP algorithm.

Both-eye recognition failure among the 3 algorithms is summarized in Table 10.

Table 10. Failed both-eye recognition.

Algorithm	Subject both eyes
Eigen-face	2,4,14,28,34
Fisher-face	2,4,28,34
LBP	1,2,4,11,14,17,29,34,35

From Table 10 above, it can be seen that the difference between the Eigen-face and the Fisher-face algorithms is in both-eye recognition of Subject 14, which is shown in Figure 20.

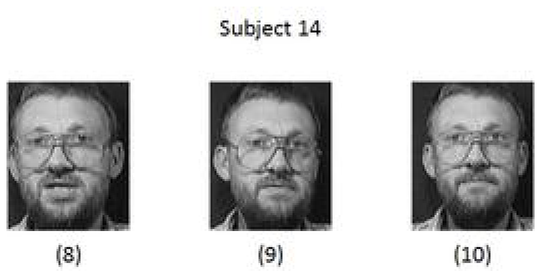


Figure 26. Subject 14 in both-eye recognition test. Subject 14 image is from the AT&T database [1].

The 3 algorithms also failed on both-eye recognition of Subject 2, 4, and 34 as shown in Figures 27, 28, and 29.

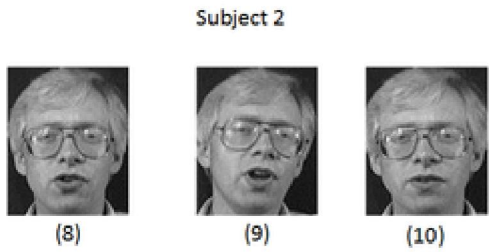


Figure 27. Subject 2 in both-eye recognition test. Subject 2 image is from the AT&T database [1].

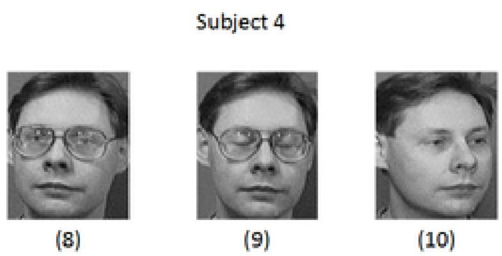


Figure 28**.** Subject 4 in both-eye recognition test. Subject 4 image is from the AT&T database [1].

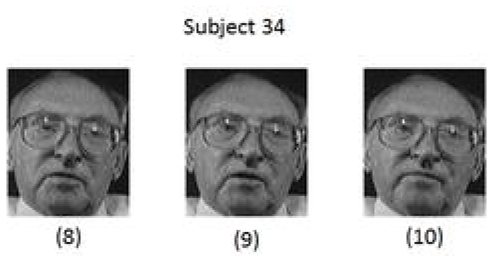


Figure 29**.** Subject 34 in both-eye recognition test. Subject 34 image is from the AT&T database [1].

From the above images, all the subjects are wearing glasses except Subject 4 (10). Note that Subject 4 (10) is in a profile pose obscuring part of the left eye. Further, in pose 9, the subject's eyes are closed under glasses. A comparison of the both-eye with the left, and the right eye recognition for the 3 Subjects 2, 4, and 34 is presented in Table 11 for Eigen-face, Table 12 for Fisher-face, and Table 13 for LBP algorithm.

Table 11. Both-eye comparison with the left and the right eye, Eigen-face.

	Subject 2	Subject 4	Subject 34
Left eye	Y	Y	Y
Right eye	Y	Y	Y
Both eye	N	N	N

Table 12. Both-eye comparison with the left and the right eye, Fisher-face.

	Subject 2	Subject 4	Subject 34
Left eye	Y	Y	Y
Right eye	Y	N	Y
Both eye	N	N	N

Table 13. Both-eye comparison with the left and the right eye, LBP.

	Subject 2	Subject 4	Subject 34
Left eye	N	N	Y
Right eye	Y	N	Y
Both eye	N	N	N

From Table 11 above, the Eigen-face algorithm recognized the left and the right eye of the 3 subjects, but it failed on the both-eye recognition for all of them. This could be attributed to the both-eye classifier and how it was trained for recognition. For example, the classifier was not trained to recognize the eye glasses.

From Table 12 above, the Fisher-face algorithm recognized the left eye of the 3 subjects but failed on the right eye recognition of Subject 4. The algorithm also failed on both-eye recognition for Subjects 2, 4, and 34.

From Table 13 above, the LBP algorithm made a positive recognition on Subject 34's left and right eyes and on Subject 2's right eye, but it failed in all other cases.

5.6 Nose distribution

Figures 30, 31, and 32 show nose distribution across 40 subjects for the Eigen-face, the Fisher-face, and the LBP algorithm.

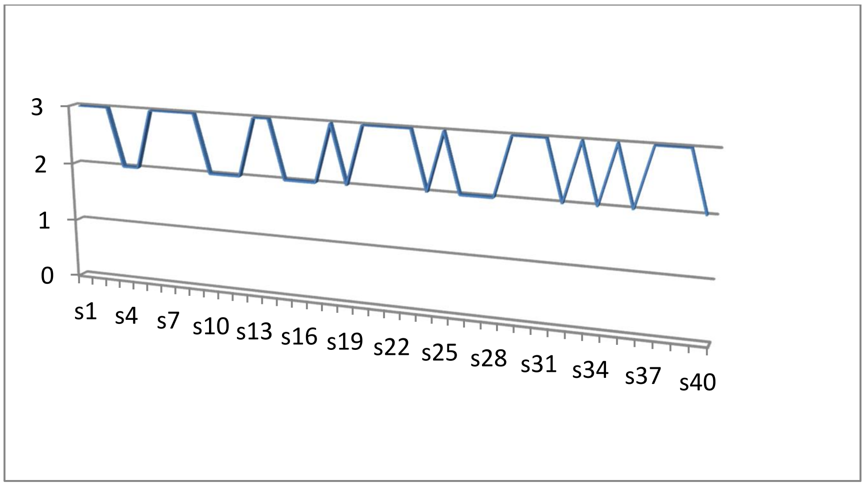


Figure 30. Nose recognition distribution, Eigen-face algorithm.

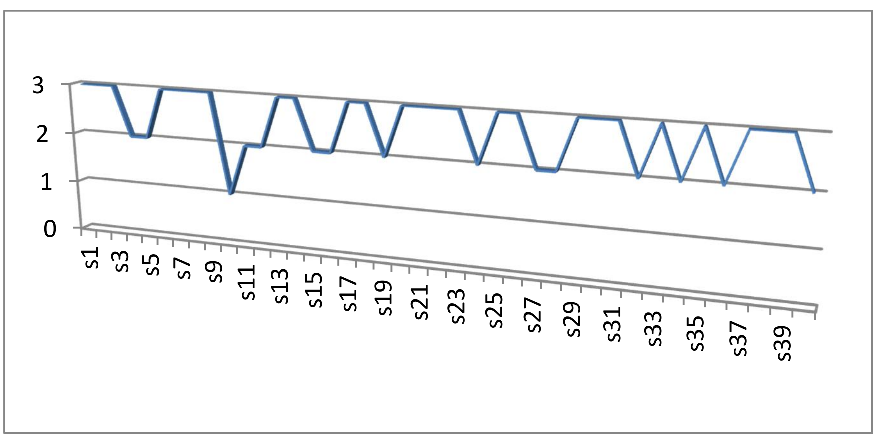


Figure 31. Nose recognition distribution, Fisher-face algorithm.

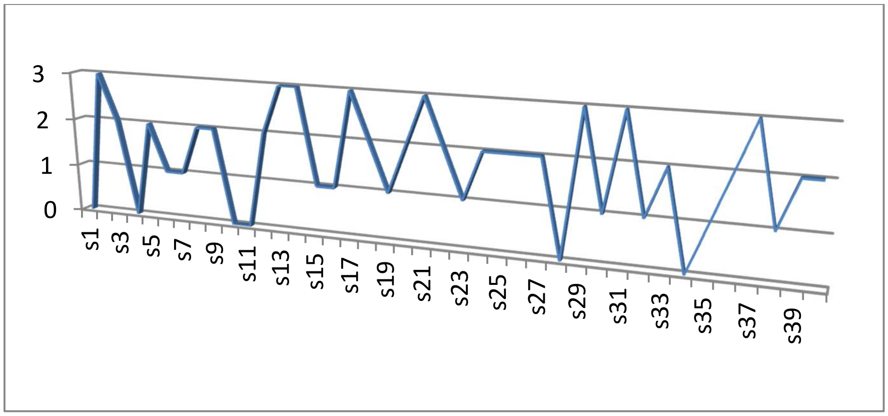


Figure 32. Nose recognition distribution, LBP algorithm.

Perhaps the most fascinating result of the study is revealed in Figure 30 and 31, where the Eigen-face and the Fisher-face nose recognition showed a 100% accuracy rate. The perfect recognition rate can be attributed to the fact that a human nose is subject to less distortion as compared to its eye or mouth. For example, a person's eye pupil can look to the left or to the right, be wide open or closed as a means of facial expression, or it can be obscured by wearing glasses. In contrast, a person's nose is usually free from such distortion under normal conditions. Since distribution of nose recognition has better overall accuracy than other features, it can be used as a dominant metric in a facial feature segmentation approach. Table 14 shows failed nose recognition for the LBP algorithm.

Table 14. Failed nose recognition.

Algorithm	Subject nose
LBP	1,4,10,11,28,34

5.7 Mouth distribution

The mouth distribution in Figures 33, 34, and 35 shows Eigen-face and Fisher-face are mostly within 2-3 recognition levels, while LBP is within a 1-2 range.

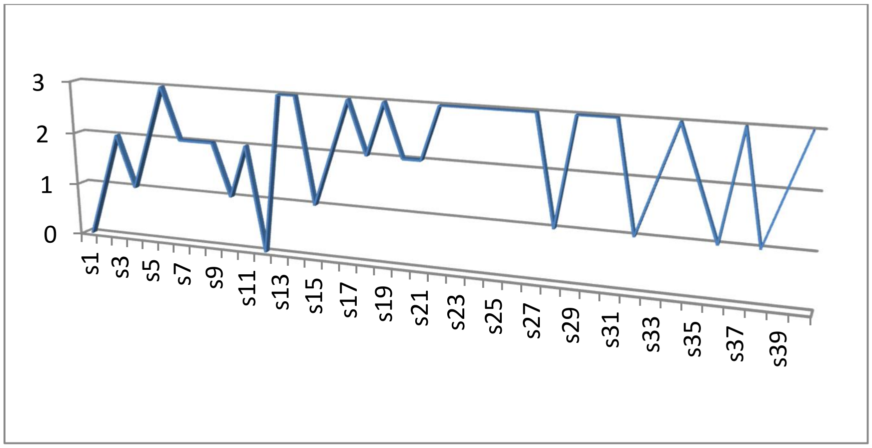


Figure 33. Mouth recognition distribution, Eigen-face algorithm.

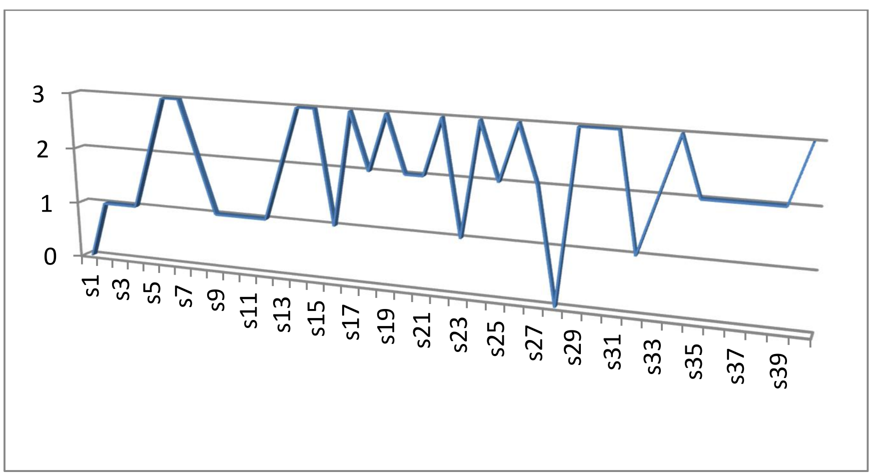


Figure 34. Mouth recognition distribution, Fisher-face algorithm.

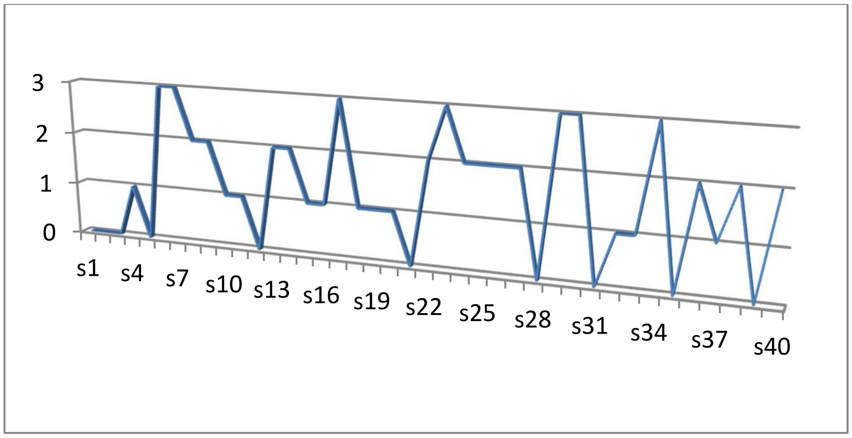


Figure 35. Mouth recognition distribution, LBP algorithm.

The results of failed mouth recognition among the 3 algorithms are summarized in Table 15.

Algorithm	Subject mouth
Eigen-face	1,12
Fisher-face	1,28
LBP	1,2,3,5,12,21,28,31,35, 39

Table 15. Failed mouth recognition.

All the 3 algorithms failed to recognize the mouth of subject 1. In addition, both Fisher-face and LBP algorithms failed to recognize the mouth of Subject 28. Figure 36 shows Subject 1, and Figure 37 shows Subject 28.

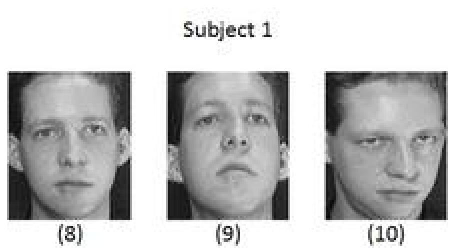


Figure 36. Subject 1 in mouth recognition test. Subject 1 image is from the AT&T database [1].

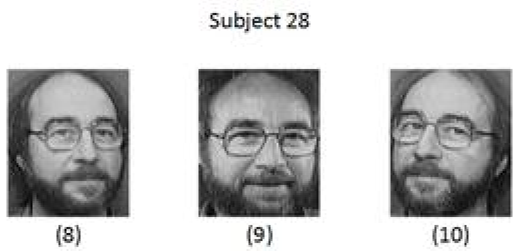


Figure 37. Subject 28 in mouth recognition test. Subject 28 image is from the AT&T database [1].

In Figure 36, Subject 1's (9) chin is up causing the mouth to be out of its normal position. A similar effect can also be observed in Subject 1 (10), whose chin is down. In either case, an out of position mouth has made it hard for the classifier to make a positive recognition. In Figure 37, the mouth of Subject 28 is obscured by a beard, causing a recognition failure for Fisher-face and LBP but not for the Eigen-face algorithm.

5.8 Algorithm

1. Prepare a "left eye" feature by running the left eye classifier on all the images in the training set.
2. Prepare a "right eye" feature by running the right eye classifier on all the images in the training set.
3. Prepare a "both-eye" feature by running both-eye classifier on all the images in the training set.
4. Prepare a "nose" feature by running the nose classifier on all the images in the training set.

5. Prepare a "mouth" feature by running the mouth classifier on all the images in the training set.
6. Train the learning algorithm for Eigen-face on each prepared feature.
7. Train the learning algorithm for Fisher-face on each prepared feature.
8. Train the learning algorithm for Local Binary Pattern on each prepared feature.
9. For a new image, prepare its feature by running the appropriate classifier.
    a. Using the Eigen-face algorithm, compare each feature of the new image with its corresponding collection in the training set.
    b. Using the Fisher-face algorithm, compare each feature of the new image with its corresponding collection in the training set.
    c. Using the Local Binary Pattern algorithm, compare each feature of the new image with its corresponding collection in the training set.
10. For a match in step 9, mark the new images as recognized. If none of the features have a match, mark the new image as unrecognized.

5.9 Results

Data for each recognition classifier is tabulated in a feature segmentation table. As shown in Table 16, an entry of 1 indicates a recognized feature, while an entry of 0 marks the feature as unrecognized.

Table 16. Feature Segmentation Fisher-face.

Subject	left eye	right eye	both eye	nose	mouth
S1_8	1	1	0	1	0
S1_9	1	0	0	1	0
S1_10	0	1	1	1	0
…..					

S40_8	1	1	1	0	1
S40_9	0	0	0	1	0
S40_10	1	1	1	1	1
Sum	84	75	70	104	90
Total	120	120	120	120	120
% recognized	70	62.5	58.3	86.6	75

In the above table, a row with all zeros is considered an error indicating none of the features of the test Subject were recognized. All such rows are then pulled out and consolidated into another table called the error table as shown in Table 17 below.

Table 17. Fisher-face Error table.

Subject	left eye	right eye	both eye	nose	mouth
s4_10	292	392	33	211	11
s5_10	213	394	217	407	393
s10_9	306	404	404	57	217
s36_9	243	196	0	363	366
	Accuracy: 96%				

The first row of Table 17 shows test Subject 4 (10), whose left eye was recognized as Subject 29 (2), and its right eye as Subject 39 (2). For the same test Subject, both-eye were recognized as Subject 3 (3), the nose as Subject 21 (1), and the mouth was identified as Subject 1 (1). Accuracy of the algorithm can be computed as the number of entries in this table over the total number of test subjects. A similar table structure is set up for LBP and Eigen-face algorithms as shown in Table 18, 19, 20, and 21.

Table 18. Feature segmentation LBP.

Subject	left eye	right eye	both eye	nose	mouth
S1_8	0	0	0	0	0

S1_9	1	0	0	0	0
S1_10	0	0	0	0	0
…..					
S40_8	1	1	1	1	1
S40_9	0	0	0	0	0
S40_10	1	0	0	1	1
Sum	57	55	81	65	58
Total	120	120	120	120	120
% recognized	47.5	45.8	67.5	54.1	48.3

Table 19. LBP Error table.

Subject	left eye	right eye	both eye	nose	mouth
S1_8	264	391	405	217	136
S1_10	0	237	233	493	66
S3_9	394	54	104	294	331
S4_10	403	211	33	233	236
S33_8	257	86	0	292	92
	Accuracy: 95%				

Table 20. Feature segmentation Eigen-face.

Subject	left eye	right eye	both eye	nose	mouth
S1_8	1	0	1	1	0
S1_9	1	1	1	1	0
S1_10	0	1	1	0	0
…..					
S40_8	1	1	1	0	1
S40_9	1	0	0	1	1
S40_10	1	1	1	1	1
Sum	83	80	82	103	81
Total	120	120	120	120	120
% recognized	69.17	66.67	68.33	85.83	67.5

Table 21. Eigen-face Error table.

Subject	left eye	right eye	both eye	nose	mouth
s5_10	266	402	403	407	193
s10_10	165	173	86	405	66
	Accuracy: 98%				

From error tables 17, 19, and 21, the Eigen-face algorithm showed the best results with a 98% accuracy rate, followed by 96% and 95% for Fisher-face and LBP respectively. The performance of the algorithms are interpreted by the number of failures they produced on each feature.

Table 22. Performance comparison of recognition algorithms based on the number of failures for each feature.

Feature	Algorithm		
	Eigen	Fisher	LBP
left eye	2	3	4
right eye	2	4	6
both eyes	4	4	10
nose	0	0	6
mouth	2	2	10

Table 22 shows that the Eigen-face algorithm performed best in all facial feature categories. A graphical representation of Table 22 is shown in Figure 38.

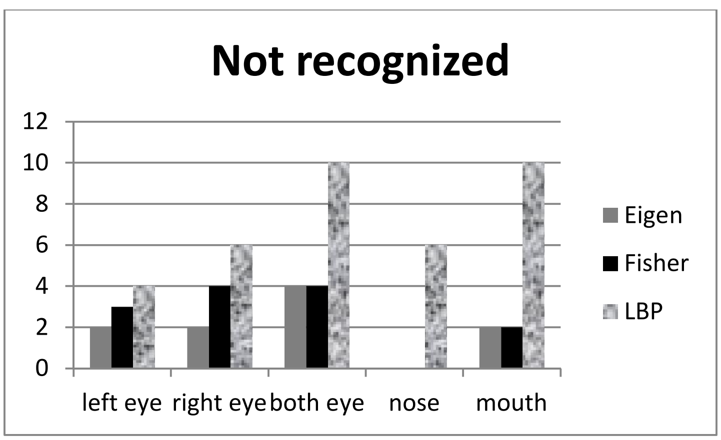


Figure 38. Summary of unrecognized features.

From Figure 38, it can be seen that the Eigen-face and the Fisher-face algorithm performed equally in the both-eye, the nose, and the mouth feature segmentation. For the left eye and the right eye, Eigen-face showed better results than Fisher-face. Note that for nose recognition, LBP had 6 failures as compared to Eigen-face and Fisher-face, which had no failure at all.

5.10 Full face vs. feature segmentation

Figure 39 shows a comparison of full face recognition with the facial feature segmentation approach (FFS). The improved accuracy in facial feature segmentation can be attributed to a finer granularity of available data. Facial features such as the nose, the left eye, the right eye, and the mouth provide more leverage to recognition strategy than full face recognition, which processes only one class of data, the whole face.

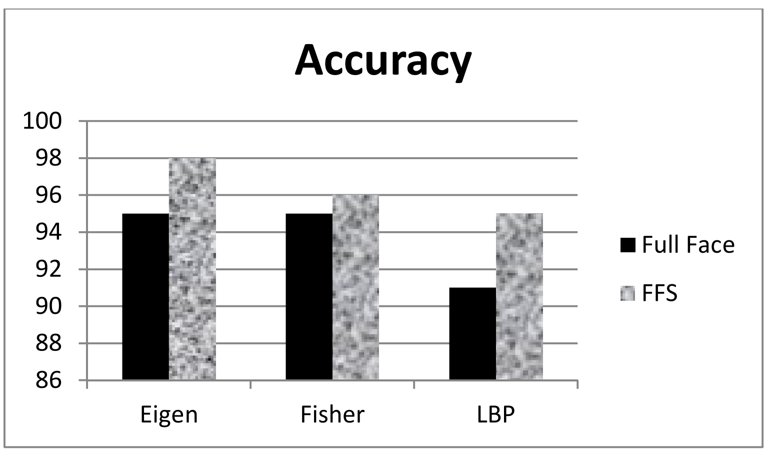


Figure 39. Accuracy of full face vs. feature segmentation.

Figure 39 shows that feature segmentation has improved the accuracy of face recognition by 3% for the Eigen-face algorithm, 1% for the Fisher-face algorithm, and by 4% for the LBP algorithm. Figure 40 shows recognition error comparison between full face and facial feature segmentation for the 120 test samples.

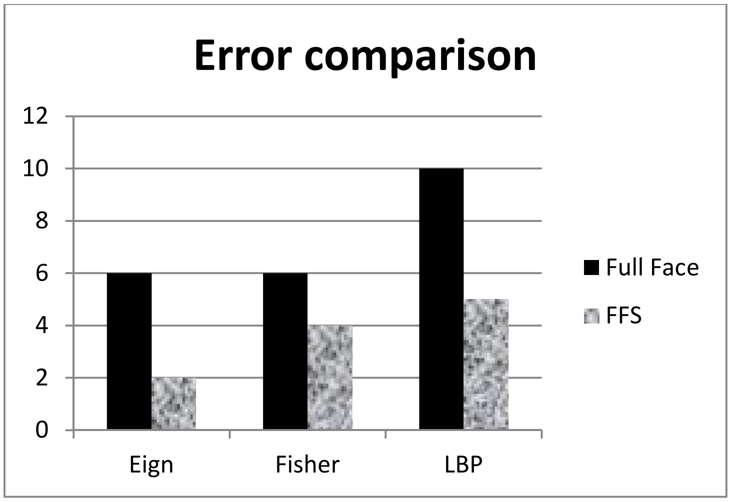


Figure 40. Error comparison full face vs. feature segmentation.

From Figure 40, it can be seen that Eigen and Fisher face algorithms produced 6 errors per 120 samples for the full face recognition approach, whereas in facial feature segmentation, they produced 2 and 4 errors for the same 120 test samples.

CHAPTER VI

CONCLUSION

Computer vision applications cover a wide range of the electromagnetic spectrum, from gamma-rays to radio waves. In contrast, face recognition, a small subset of computer vision, is limited to the narrow band of visible light. A major factor affecting recognition algorithms is the illumination and the intensity of visible light. One popular method of reducing this effect is the histogram equalization, where the intensity of light is distributed evenly across the entire image. The Eigen-face and the Fisher-face algorithms are appearance based, where image pixel data are used as a whole to perform recognition. Both algorithms use the subspace projection for comparing images by calculating image separation in a reduced dimension space. The method of reducing dimensions for the Eigen-face algorithm is PCA and for the Fisher-face algorithm is LDA. Reducing dimensions is an important technique for eliminating noise and improving recognition accuracy. Local binary pattern is a textured-based algorithm. It regionalizes an image, so that for each region, a center pixel is used as a threshold to compute a new value for its replacement. A histogram of all regions is assembled and combined to represent the LBP image. The Eigen-face and the Fisher-face algorithms take a holistic approach to face recognition by looking at the whole image as a high dimension vector and then applying PCA or LDA for dimension reduction. The LBP algorithm, on the other hand, looks at local features, where dimension reduction is implicitly applied. Using AT&T data set for both training and testing showed a 95% accuracy rate for Eigen-face and Fisher-face and a 91% accuracy rate for the LBP algorithm. The results of applying facial feature segmentation method to the same data set and algorithms showed accuracy rates of 98%, 96%, and 95% to the face recognition. The improvement in accuracy is attributed to finer granularity of data, since recognition is performed on a number of facial features, including the left eye, the right eye, the nose, and the mouth. In a holistic approach, a failed recognition has no recourse strategy, but in facial segmentation, feature sets provide more opportunity for a positive recognition outcome. Facial feature segmentation also reduces the effect of light since smaller regions of the face are selected. In this case, algorithms that are susceptible to the effect of light will produce more accurate results since they work on a smaller region of the face, where the effect of light may be absent or less dominant. The accuracy rate can be further improved by manipulating an image's pixel or its orientation. For example, if the left eye is not detected, the image can be shifted up or down to move the eye within the detected range. If a face is in profile orientation, it can be rotated to bring the face into portrait orientation, allowing more facial features to be detected. Although image pre-processing can enhance the accuracy of facial recognition algorithms, no attempt was made to pre-process images in an effort to keep the outcome of the results free from any image pre-processing influence.

CHAPTER VII

CURRENT AND FUTURE WORK

The study presented in this thesis is the first of a three part project. Here, the objectives were to improve the accuracy of classical face recognition algorithms in a non-mathematical approach. A face recognition system loaded with many such algorithms and operating on 2D images can choose a recognition strategy that yields the best results. The development of facial feature segmentation and its associated tables as described in section 5.9 provides the tool for such systems. Recently, employing 3D images in face recognition has risen in popularity and many impressive results have been published in various academic and scientific journals. In 2014, researchers at Facebook published a paper called "Deep Face," describing a nine layer neural network, using explicit 3D modeling of 2D images with a 97.3% recognition accuracy rate. The second part of this project is an extension of "Deep Face," currently under development by the author of this thesis. The neural network portion of the project has been completed and tested on the AT&T data set. For 3D models, "Deep Face" rotates and aligns 2D images around 6 fiducial points, which are then wrapped around a 3D image plane using a 3D affine camera. Our method of generating a 3D image is based on the Active Appearance Model, where a generic deformable model is built around 67 landmarks of a 2D image. The generic model is then fitted to a Delany Triangle computed from the landmarks of a particular image to render its 3D equivalent. The third part of the project is designed to use face recognition in a deep neural network using GPU clusters. One of the objectives in this part of the study is to examine the performance of recognition algorithms as the volume of data increases using MPI (message passing interface) in a parallel computing environment. Another objective is to implement an automated feature extraction that helps in deciding the best way to use face data for recognition. In addition, further topics to be examined include scalability issues in high throughput neural networks and machine learning instead of parallel computing.

References

[1] *AT&T data set* (n.d.) [Online]. Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/pub/data

[2] C. H. Chen, *Computer Vision in Medical Imaging*. vol. 2, University of Massachusetts, 2014.

[3] *Computer Vision Technology for Food Quality Evaluation*, Dec 2007 [Online]. Available: http://www.researchandmarkets.com/reports/1758859

[4] *Electromagnetic spectrum,* (n.d.) [Online]. Available: http://www.darvill.clara.net/emag/

[5] *Biometrics*, (n.d.) [Online]. Available: http://en.wikipedia.org/wiki/Biometrics

[6] P. Viola and M.J. Jones, "Rapid object detection using a boosted cascade of simple Features," *IEEE Transactions on Computer Vision and Pattern Recognition*, vol. 1, pp. 511-518, 2001.

[7] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," *IEEE Transactions on International Conference on Image Processing*, vol. 1, pp. 900-903, 2002.

[8] *Yale Face database* (n.d.) [Online]. http://vision.ucsd.edu/content/yale-face-database

[9] R. Brunelli and T. Poggio, "Face recognition features vs templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp 1042-1053 Oct 1993.

[10] A.V. Nefian. "Statistical approaches to face recognition." PhD Thesis, Georgia Institute of Technology. Dec 1996.

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

[12] R.N. Belhumeur, J.P. Hespanha and D.J. Kriegman, "eigenfaces vs fisherfaces Recognition using class specific linear projection," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, July 1997.

[13] T. F. Cootes, K. Walker and C. J. Taylor, "View based active appearance models*,"* *IEEE International Conference on Automatic Face and Gesture Recognition*, *Grenoble, France*, pp. 227-23, Mar 2000.

[14] 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, Dec 2003.

[15] G. Guo, S. Z. Li and K. Chan, “Face recognition by support vector machines,” *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 196-201, Mar 2000.

[16] B. Moghaddam, T. Jebara and A. Pentland, “Baysian face recognition,” in *Pattern Recognition*, vol. 33, no. 11, pp. 1771-1782, Nov 2000.

[17] S. Lawrence, C. L. Giles, A. C. Tsoi and A. D. Back, “Face recognition: a convolutional neural network approach,” *IEEE Transactions on Neural Network*, vol. 8, no. 1, pp. 98-113, Jan 1997.

[18] T. Ahonen, A. Hadid and M. Pietikanen, “Face description with local binary patterns: application to face recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037-2041, Dec 2006.

[19] P. Penev and J. Atick, “Local feature analysis: a general statistical theory for object representation,” in *Network: Computation in Neural Systems*, vol. 7, no. 3, pp. 477-500, Oct 1996.

[20] W. S. Yambor. “Analysis of PCA-based and fisher discriminant-based image recognition algorithms.” M.S. Thesis, Technical Report CS-00-103, Computer Science Department, Colorado State University, July 2000.

[21] G. Shakhanrovich and B. Moghaddam, “Face recognition in subspaces,” in *Handbook of Face Recognition,* Eds. Stan Z. Li and Anil K. Jain, London: Springer-Verlag. 2004, pp. 19-49.

[22] J. Zhang, Y. Yan and M. Lades. “Face recognition: eigen face, elastic matching and neural nets,” *Proc. of IEEE* vol. 85, no. 9, Sep 1997.

[23] R. C. Gonzalez and R.E. Woods, *Digital Image Processing (2rd Edition)*. Boston, MA: Addison-Wesley Publishing Company 2001.

[24] R.A Fisher, “The use of multiple measures in taxonomic problems,” *Annals of Eugenics*, vol. 7, no. 2, pp. 179-88, Sep 1936.

[25] A.M Martinez and A.C Kak, “PCA versus LDA,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, no. 2, pp. 228-223, Feb 2001.

[26] C.H. Chen, L.F. Pau and P.S.P Wang. *The handbook of pattern recognition and computer vision (2nd Edition)*. Singapore: World Scientific Publishing Co, 1998, pp. 207-248.

[27] D. He and L. Wang, "Texture unit, texture spectrum and texture analysis," *IEEE Transactions on Geosciences and Remote Sensing*, vol. 28, no. 4, pp. 509-512, Jul 1990.

[28] T. Ojala , M. Pietikaien and D. Harwood, "A comparative study of texture measures with classification based on feature distribution," *Pattern Recognition* vol. 29, no. 1, pp. 51-59, Jan 1996.

[29] T. Ahonen, A. Hadid and M. Pietikainen. "Face recognition with local binary patterns," in *Proc. ECCV*, 2004, pp. 469–481.

[30] M. Akhloufi and A. Bendada, "Locally adaptive texture features for multispectral face recognition," *IEEE International Conference on Systems Man and Cybernetics (SMC)*, pp. 3308-314, Oct. 2010.

[31] C.H. Chan, J. Kittler and K. Messer. "Multi-scale local binary pattern histograms for face recognition," in *International Conferences on Biometrics*, 2007, pp. 809-818.

[32] R. Brunelli and T. Poggio. "Face recognition through geometrical features," in *Proc. ECCV*, 1992, pp. 792-800.