ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

REAL-TIME FACE RECOGNITION USING EIGENFACES

Uzair Saeed*, Engr Asad Ullah, Zohaib Latif

* Assistant Professor (Department of Computing),Riphah International University, Faisalabad
Lecturer (Department of Electrical Engineering),Riphah International University, Faisalabad
Department: School of Computer Science and Technology University: Beijing Institute of Technology, China

DOI: 10.5281/zenodo.160868

ABSTRACT

Access control systems are used to grant or deny the access to a person of a particular resource. There has been an enormous change in the trend of access control systems in recent times. Starting with the use of physical access control systems such as tokens, passwords etc., for the identification of a person, the trend has swayed towards designing and deploying of access control systems which use biometric identification of individual persons, for the grant or denial of access to resources.

Biometric identification methods use various sources like retina, fingerprint, DNA etc., Biometric sources can be classified into two, namely physiological and behavioral. The former includes face, fingerprint, hand, iris, DNA and the latter includes keystroke, signature and voice. The access control systems using these biometric sources fundamentally identify and recognize a particular personal trait of person and compare it with the information available to grant or deny access to such person who seeks to interact with such system. As amongst such biometric sources to develop reliable access control systems researches have shown overt interest in using the face of a person (face recognition). Such inclination of the researchers is due to the various strategic advantages face recognition systems have like, its global application, wide and compatible collectability of data, cost effectiveness in implementation (for example existing surveillance cameras can be used to deploy such systems) when compared to other biometric methods and many more.

The current project fundamentally aims at successfully designing and implementing a face recognition technology to develop an access control system. Out of the various available methods for developing a face recognition technology such as Fishersface, Hidden Markov model, dynamic link matching, three-dimensional face recognition, Eigenfaces etc., this project adopts Eigenfaces method of face recognition to achieve such aim.

The project fundamentally detects and identifies human faces that work as a biometric source. This project aims to provide solution for the development of face recognition system by assuming that, problems involved in developing such systems are intrinsically a two dimensional rather than a three dimensional. It basically identifies the face of a person in a face image and then identifies the specific characteristics of such face image, then compares such characteristics with an existing database containing specific characteristics of several faces of different individuals, to decide if the former matches with any of the existing faces in the database. Eigenfaces method is utilised to achieve the above result where, face image is projected onto a feature space that spans the significant variations among known faces. The best variation among different images is calculated where during such calculation not exactly the facial features like eyes, nose, ears, etc., are classified. Instead it learns each face in an under constant observation as a whole.

INTRODUCTION

In this modern world, where the trend changes in every field, people always look for highly sophisticated technology to access each of their secure personal belongings. But still, large numbers of people use access control like PIN, keys, passwords, identity cards. These methods are highly secure but still are vulnerable because they could be easily forgotten or lost. When, considering information system, it is been attacked by numerous hackers by simply obtaining others password.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Safety and security have always been of high importance for business establishments, banking and other industrial and academic institutions. To meet the ever increasing security and safety requirements public places like airports, stadiums, shopping malls also use various types of recognition systems. Various physical access control and electronic access control mechanisms have been used to achieve this task with highly preferred being electronic access control. Increasing number of business and other establishments are now adapting to these high tech technologies as this could cut the number of manual labour and its cost, less cost in implementation and also saves time. Access control can be used to permit or deny the use of particular resource by a particular entity e.g. a bank account, where the account holder is authorised to withdraw the money. Electronic access control uses credentials to grant the access. In broad sense credential can be something you have, something you know, something you are or any combination of these e.g. a swipe card, a PIN number or a biometric feature.

Creating a real-time face recognition system is difficult, because faces are more meaningful, complex, and multidimensional [1]. And also, face of an single individual gives variations during different timings, such that, appearance of same face changes because of ageing, facial expression, illumination, different view point, facial hair, use of other accessories like sun glasses etc., [2]

There are two different approaches for face recognition, feature based approach and template matching approach. In feature based approach a face is carefully noted far the features of the face such as eyes, nose, ears etc., and these facial features are then computed to describe a face. Then, they are compared with the facial feature stored of several other faces. This method is quiet computationally expensive, as each of the face features that describes a face is computed separately, also requires careful selection of facial features. In template matching face images is analysed as whole rather than separating its parts. During this process the image is read pixel-by-pixel to find the characteristic of a face and then compared with other known faces in the database. One of the methods that give promising result using this approach is PCA (Principal Component Analysis), where face is defined by subsets of eigenvectors. Therefore, it is called Eigenfaces [3].

In this method considering 2-D picture the full model of the face is not considered, such is the detailed geometry of having three-dimensional characteristics. And the aim is to keep it simple as possible. Even though the face is complex, we take advantage of the structural impose of the picture. Where the face image is transformed set of small facial characteristic, consist of “Eigenfaces”. And then the new faces are projected to the face-space and comparing its position to other individuals of known faces.

This method has a major advantage over other method in terms of speed and simplicity.

LITERATURE REVIEW

Algorithms Based on Image-base Face Recognition

Image based face recognition is divided into two types they are Appearance-base and Model-based. Their classification and explanations are shown as follows

Appearance-based
Linear analysis: PCA, ICA, LDA
Non-linear analysis: KPCA, ISOMAP

Model-based
Elastic Bunching Graphing
2-D Morph able
3-D Morph able

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

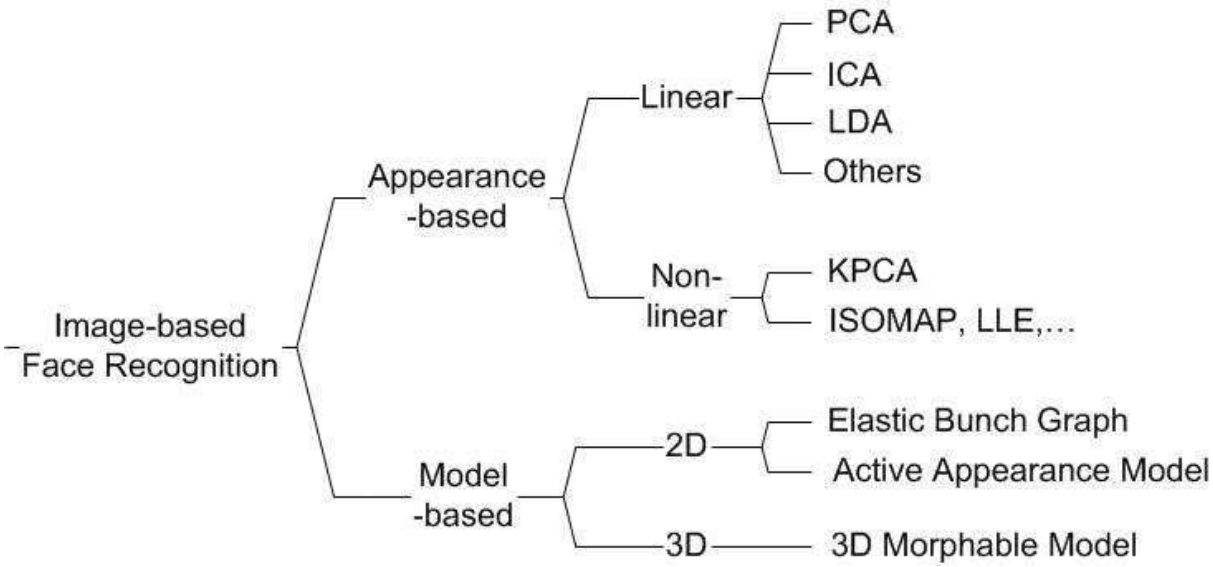


Figure shows simple classification for image-based face recognition [4]

Algorithms Used for Appearance-based Face Recognition

The appearance based face recognition are generally divided into two part, they are linear and non-linear types as shown in the above figure.

Linear Analysis: Linear analysis can be simply explained as, the face vector for an image obtained are projected to the basic vectors. The input image of high dimensional space can be reduced to lower dimensional space, thus reducing the original dimensionality of input image. Matching between the input image and the training set of image is calculated by the difference between the vectors projected. The more score between matching, represents more similarities between images.

Some of the popular methods of linear analysis are PCA, LDA and ICA is explained below.

Non-linear Analysis: Linear analysis is insensitive to relationship between multiple pixels in an image. The non-linear relation for a face exists in a complicated view point of a face, different illumination and facial different facial expressions. Methods like Kernel PCA, Kernel ICA etc., are used to extract these non-linear features of a face. [5]

Principal Component Analysis

When considering to dimensionality deduction, one of the widely used method is based upon Principal Component Analysis method (PCA). This method is used to extract the features so that the recognition problem could be solved. [6]

This statistical dimension-reduction method produces the optimal linear least squares decompositions of a training set. In this method input is a training set, t1,….., tN of N facial images, so that the mean of the facial training set is zero(Σti=0). Each image is interpreted as a point in N^ (m*n), where m and n are pixels in x and y axis of the image. PCA finds optimal linear least-squares in (N-1) dimension space. The PCA is characterized by set of eigenvectors (e1,….N-1) and eigenvalues (1,…..N-1). The eigenvectors obtained are referred to as Eigenface. These Eigenface consists of the calculated free space which compared to the calculated index of sample images. Finally the image which has the greatest point in matching to the Eigenface is declared image match. [7]

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

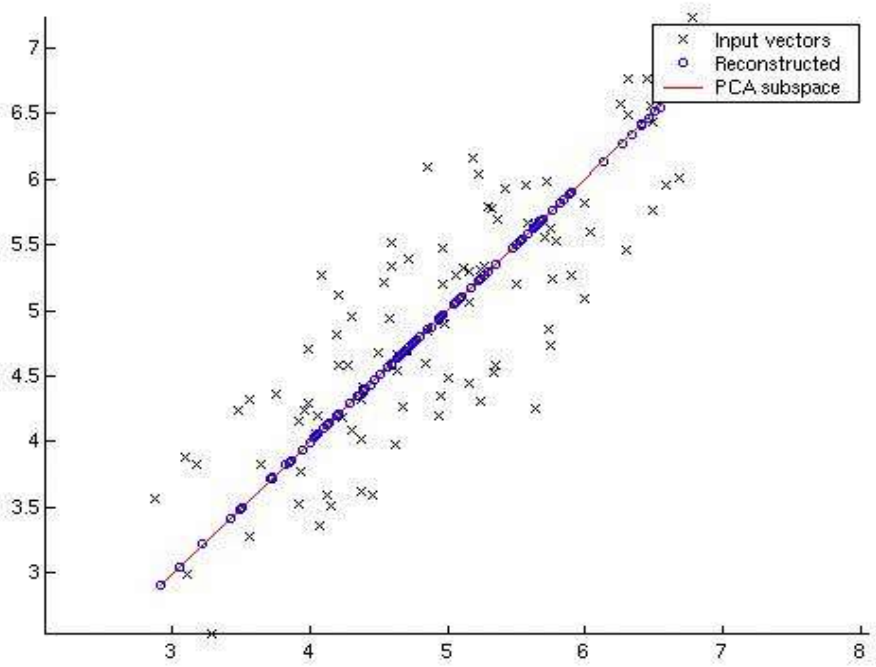


Figure: Linearization of classes using PCA [8]

Independent Component Analysis

ICA (Independent Component Analysis), in which the Independent Components (ICs) obtained by projecting the image onto the subspace spanned by the base. By this better intrinsic properties and local characteristic of the facial data set can be obtained [9]. In simple words, ICA removes high-order statistical dependencies, therefore producing spares and independent code for pattern computation [10].

In a particular image X,

$X \epsilon R^N$ where N is the dimension of the image

By normalising the image with rows and columns, the covalent matrix given as

$\Sigma x = E \{[X-E(X)] [X-E(X)^T]\}$ where E is expectation operator, T is transpose operation

The above equation can be written as

$\Sigma x = F \Delta F^T$

where Δ is diagonal real positive and F transforms original data X into Z.

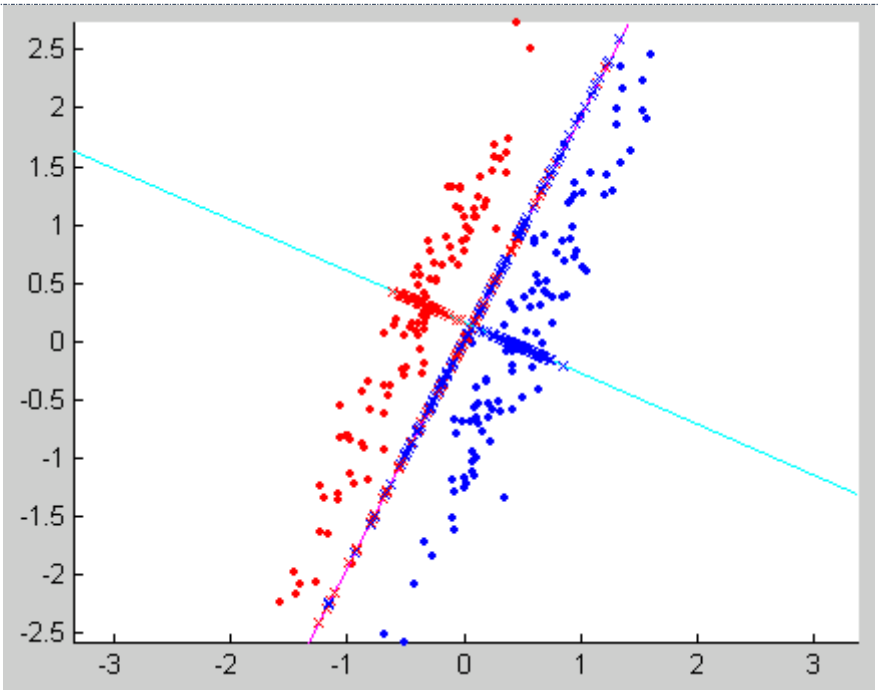
$X = FZ$

Therefore new data F is independent or "the most independent possible".[11, 12]

Linear Discriminated Analysis

Linear Discriminate Analysis is also called as Fisher's method (Fisher (1936) – introduced two-class classification). LDA is classical statistical technique which produces data of classes according to the patterns. They form closer classes for same data and different classes for different data [13]. Even though LDA method is successful in many application and been a useful tool for pattern recognition, it has a disadvantage known "small sample size problem" (SSS) [14], [15].

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7



In above figure the purple line is PCA and the crayon is the vector obtained by LDA [16].

Comparing the two projections from the figure, PCA smears the classes together so that they are no longer spreadable. On the other hand LDA achieves better separation of classes and therefore classification becomes easier [17].

Kernel Method

Kernel Principal Component Analysis method is an extension of method like PCA, ICA etc., [18]. In this modelling technique in the input the input vector is projected to a high dimensional linear space created by Principal Component Analysis method. Therefore, non-linear components are identified using Kernel function K(x, y), where kernel K is given by

$$K(x, y) = ((x^T) * y + 1)^2$$

and Gaussian kernel is given by

$$K(x,z) = e^{-\|x-z\|^2/2\sigma}$$

The above kernels are used to obtain the non-linear components of image [19].

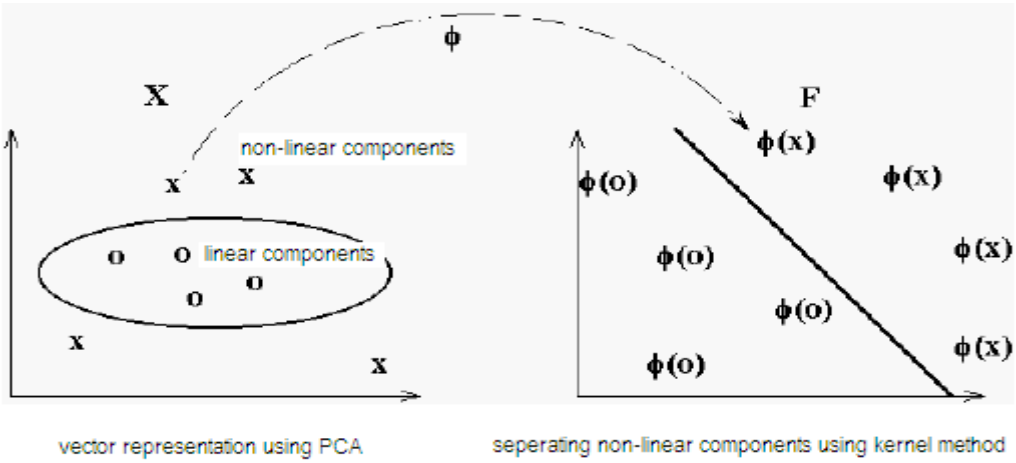


Figure: Separation of non-linear components using kernel method [20]

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Model-based Face Recognition

Model-based face recognition can be explained as to construct a model of a human face for recognition. The constructed model face model is then used to compare the variations with that of other faces. In this, model-based approach we derive the relative position of the features and their distance form the facial elements like eye, nose etc., in more generally, this method consist of detailed information of a face and the texture of the face.
Model-base face recognition is divided into two types

- **2-D modelling** consists of Elastic Bunch Graphing and Active Appearance Model.
- **3-D modelling** consists of 3-D Morph able modelling.

Elastic Bunch Graph Matching (EBGM)

Considering, face images are represented as a same or topographical in nature. So, a graph of a face is produced from the face structure. This graph created by different node points and edges labelled with distance vector and this graph is called Face Bunch Graph (FBG). Each node on this graph holds bunches of description about the facial features of the face. A simple representation of EBGM is shown below

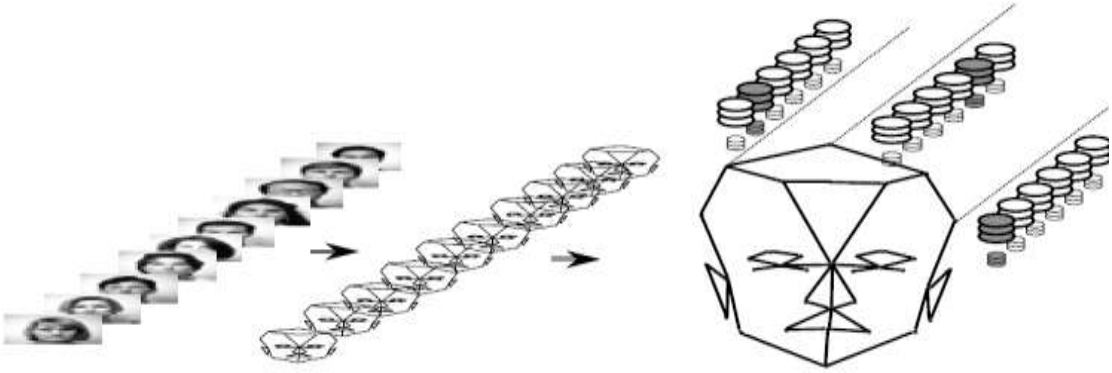


Figure: Face Bunch Graph (FBG) [21]

Each of the nodes consists of number of disks. These disks represent the description of facial features. Each of these disks is called jet representation [22].

A bunch graph is created for each of the nodes for all the known faces. This gives a bunch of model jets for the known faces.

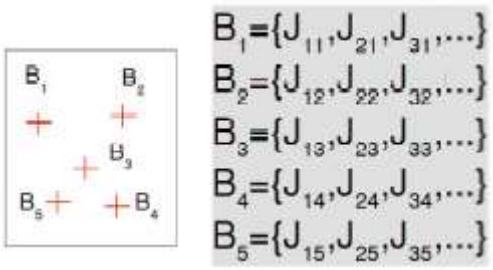


Figure: Jets extracted for known faces [23]

These jets are compared with the input image node points. Thus, matching images will contain similar jet values [23].

Active Appearance Model (AAM)

Active Appearance Model is based upon set of labelled images. In each example face, a landmark point is marked. The landmark point in a key position describes the facial features of a face. This method combines with the linear analysis method, so that, the vector obtained are combined to control the shape and texture of the created model. Thus, comparing these shape and texture, image matching is performed.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

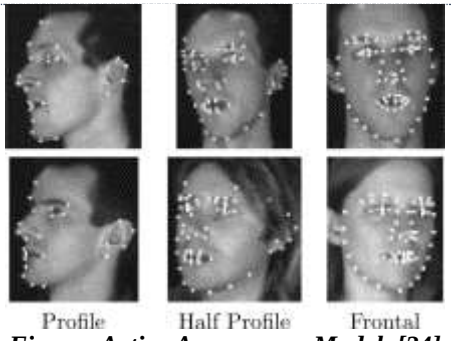


Figure: Active Appearance Model [24]

The shape of the model is controlled by x vector and texture of the model is represented by g, c is the parameter which controls the shape and texture of the image.

Therefore, the obtained values are represented as follows

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Q}_s\mathbf{c} \quad \mathbf{g} = \bar{\mathbf{g}} + \mathbf{Q}_g\mathbf{c}$$

Where, $\bar{\mathbf{x}}$ is the mean of shape and $\bar{\mathbf{g}}$ is the mean of texture. Q_g and Q_s are created from the eigenvectors of a covariance matrix [24].

3-D Morphable Method

In referring to the past decades, most of the pattern recognition has been focused on 2D images [25]. Considering 2D images which face difficulties like variation of poses, expression, illumination and ageing. And it's been reported by face recognition vendor test (FRVT) [26].

As human face, is lying in a surface of 3-D space. So, the method of 3-D face recognition is considered as more suitable for representing a face. This is because it has a strong ability to reduce the problems like head pose and illumination.

In this component-based system in which the training set of images needed in a large number under different lighting conditions. And based upon three images of a person, the morphable model allows the computation of 3D using synthesis method [27].

Figure: Example of input image obtained

Figure: 3-D models generated from obtained input [28]

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Video-based Face Recognition

During the past few years more and more research has been done from image sequences. This is been difficult because low quality of images and the images are small [29], [30]. Using, sequential importance sampling (SIS) [31-33]. By this method in which the stilled image obtained from the video samples, which represents individual images thus applying any of the methods like Fishersface, PCA, 3-D modelling etc., to do the normal face recognition process.

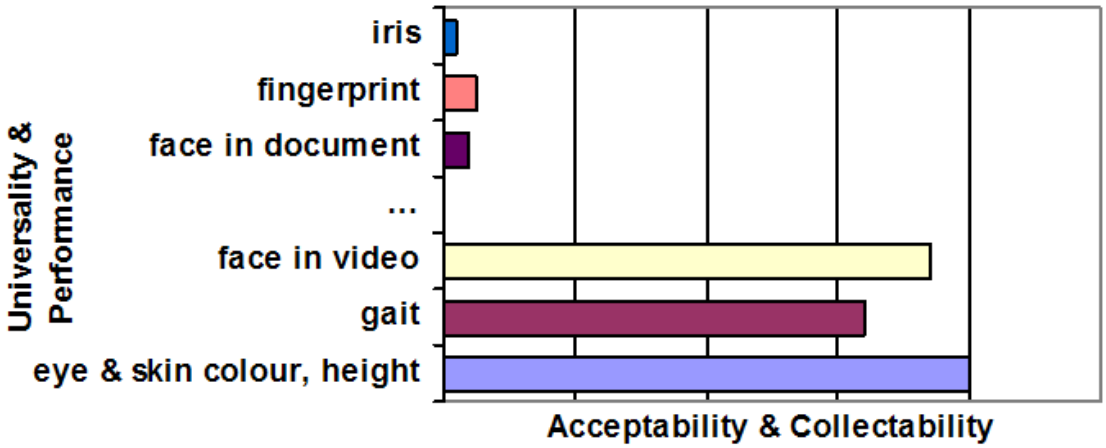


Figure: Quality vs. availability of different image-based biometric modalities [34]

There are also research based on face recognition in video (FRiV), which dose not divide video into images, but considering the whole video instead [34].

Market Research

The product to be studied is a Face Recognition system. They are security systems which are computer based and are capable of detecting and identifying human faces (http://epic.org/privacy/facerecognition/). The face recognition system uses biometrics which uses a computer driven technology and identifies the face of the individual by comparing it with its digital image. The demand for this product has been increasing steadily over time and after the unsuccessful results obtained from the use of CCTV cameras propelled the use of Face Recognition systems on a wider scale. New technologies are helping in this product making the necessary advancements and make it even widely acceptable across a wide forum. This topic can be further understood with the help of a case study that was undertaken which is as follows:

Case Study: Aurora Computer Services Ltd

The sketchy pictures that were obtained with the help of CCTV or recording by any other means meant that it blocked all the possible methods of future investigations. There are cases where the police force has the database of hundreds and thousands of photographs, and very often the offenders are people who commit repeated crimes and are in the list of the police but it is practically impossible to compare and trace a single photograph from the list. The product developed in this case, was software which could differentiate and compare from around 200,000 records of offenders and point out towards the criminal concerned with the help of a digitised image. The search can be completed within a second. This Aurora system is used by various police forces around the UK to help increase the conviction rate.

The above shown is an example of how the software works and its applications. (Source: www.auroracs.co.uk).

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Market Overview

The security concerns which have been growing wider and wider following global terror attacks, the governments across the world have been steeping up security measure. USA is the leading market to adopt and implement face recognition system. Industry risk rating report states that the society we are living in would be one in which every action or deed cannot go unnoticed. This system brought more accountability or greater checks on American citizen, as even car rental companies had installed satellite or navigation systems in cars to trace or find whereabouts of driver.

These situations thus demand more sophisticated and better security solutions than ever before. With this is mind, the facial recognition system with biometrics are thus expected to notch an compound annual growth rate (CAGR) to 27.5% from the year 2005 to 2012. North America is said to be leading in terms of use of these systems although the Middle East and European regions would also contribute to the growth in this region.

With the advancements in technology, the prices of these products could fall making them affordable for a wider customer base. The face recognition technology is also suitable for a variety of niche products, thus creating immense opportunities for these products. These advantages make it eminent for the technology to attain high growth levels.

Products that exist in the market currently with similar technologies would thus be studied which are as follows:

Facial recognition Surveillance System:

Figure: Facial Recognition Surveillance System

The above figure shows the schematic of a Facial Recognition Survelliance system. **Fx Alert** is a security system which is computer based, which is capable of automatically detecting and identifying faces of human beings. It uses one of the latest technologies for face recognition. In real time, List matching engines are showed with the help of this system.

Features:

- It is a robust technology which can accommodate minor variations in lighting and pose for high accuracy.
- It performs the facial identification at the rate of 750,000 records/ second with a great degree of accuracy.
- It can accommodate several cameras and many watch list items due to its scalability.
- It can identify and verify multiple faces different faces in crowded places in real time.
- It can also easily integrate with the video surveillance system that currently exists.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

First Facial recognition

Figure: First Facial Recognition

The above figure is a schematic of the First facial recognition system. Fx Guard is known to be the first facial recognition software that mainly operates on the TCP/IP protocol. It functions with the help of the latest and a very precise facial recognition engine.

Features:

- The process of facial verification is done in less than a second.
- It can integrate quite easily with the corporate LAN and is also compatible with personal computers.
- It works on a simple plug and play appliance for time, attendance and access control applications.
- It is configured with the help of a built in Web server and the reports can be viewed with the help of web browsers available.
- It has an integrated infrared sensor and an RFID reader.
- There is no need to install any additional software when used on personal computers and it also provides cross platform compatibility.

Face Recognition system

Figure: Face Recognition System

The above shown figure is a schematic of the Face Recognition system. The features for this system are mentioned as follows.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Features:

- The video camera captures the face of every person at the point of entry and if the facial features match with the stored features only then access is permitted.
- In areas where security required is high, it can be combined with the terminals of the card so that the owner can use the individual card.
- Terminals which have entries can be networked together to enable the circulation of face data to all terminals after central enrolment is done.
- The security personnel can be provided with added visual control.
- The facial images which are captured can be stored in the log so that they can be inspected later with the help of standard browsers.
- The recognition time is less than a second.
- It is capable of providing auditing and reporting, it can also provide live checks so that it can be protected against major frauds.

AIM OF THE PROJECT

Aim of the project: This project aims to develop a face recognition system which recognises a two dimensional image (face image). This is done by using EIGENFACES[1] method. For such purposes this project is fundamentally divided into three main stages. They are project conception, planning and execution.

Project Conception: This stage is primarily aimed to understand and analyse the fundamental theory supporting this project. During this stage the various mathematical methods underlying the theory are studied and detailed along with the study of various functional components included in the project.

Project Planning: This stage primarily elucidates the methodology adopted for planning and execution of the project. This stage further focuses on designing and analysis of the project. The same is conducted by using two tools. They are 1.SELECT Yourdon 2 IBM Rational Rose.

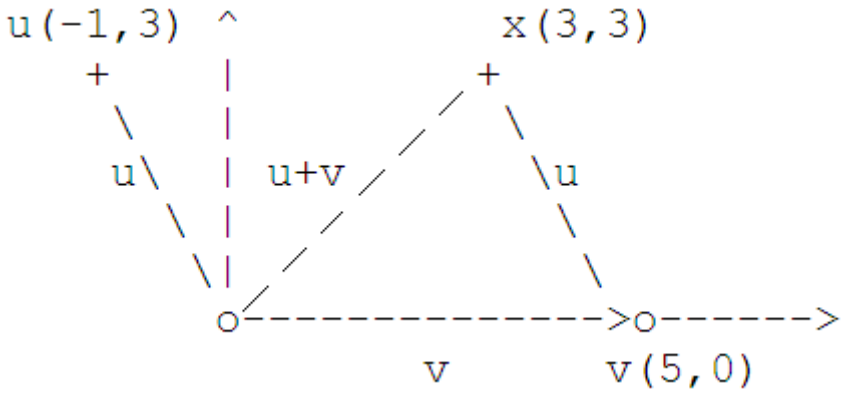
Project Execution: This stage included writing, testing and implementation of the program/software.

PROJECT CONCEPTION

Explanation of Fundamental Mathematics Involved

In this part we would review the basic of mathematics that's involved in this project.

Vector: This can be simply described by quantities which are fully described by both magnitude and direction [35].



In the above example for a vector shown in which vectors v = (5, 0), u = (-1, 3) and the vector x = (3, 3) which is the sum of vectors u + v = x.

[1] Eigenfaces was developed by Sirovich and Kirby (1987) and used by Matthew Turk and Alex Pentland in face classification .

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Matrix: A matrix can be represented as M = (m × n). Where m is the number of rows and n is the number of columns. Further details about matrix's [36]

Eigenvalues and Eigenvectors:
System of linear equation given in mathematical terms as follows

$$a\,x = b \qquad \text{equ. 1}$$

Where a and b are known numbers, while x is unknown numbers, this is called **linear equation** [37].

The equation given for eigenvalues and eigenvector are

$$A\,x = \lambda\,x \qquad \text{equ. 2}$$

Where A is the square matrix, x is called the eigenvector and λ is the eigenvalue [37].

For example:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} X1 \\ X2 \end{pmatrix} = \lambda \begin{pmatrix} X1 \\ X2 \end{pmatrix}$$

From the above equation we get the eigenvalues λ1 = X2 and λ2 = -X2

These eigenvalues can be represented in vector space X as shown

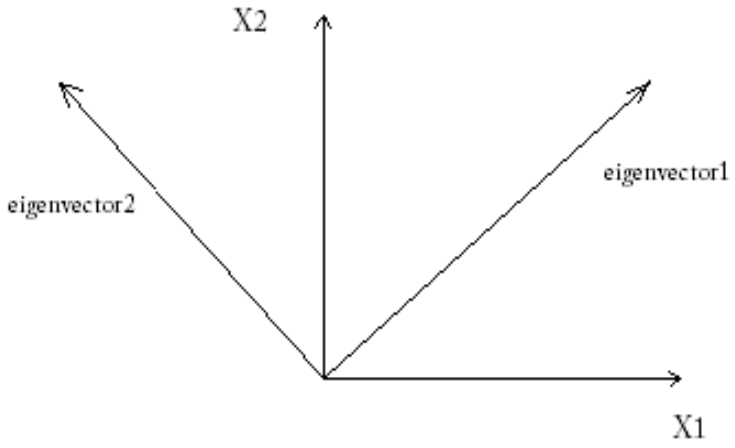


Figure: Representation of eigenvector

Calculating Eigenvectors:
From the equation 2 above we get

$$(A - \lambda\,I)V = 0 \qquad \text{equ. 3}$$

Where I is the identity matrix (n × n) and V is eigenvector. The above equation is a homogeneous system, therefore solution exists if only [38]

$$\text{Det}(A - \text{eigenvalue}\ I) = 0 \qquad \text{equ. 4}$$

For this above equation ***3*** considering an example to find eigenvalues

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

$$\begin{pmatrix} 3 & 1 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} v1 \\ v2 \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \begin{pmatrix} v1 \\ v2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Solving the above equation we get

$$\begin{pmatrix} 3-\lambda & 1 \\ 2 & 2-\lambda \end{pmatrix} \begin{pmatrix} v1 \\ v2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

By equation ***4*** solutions exists if

$$\begin{pmatrix} 3-\lambda & 1 \\ 2 & 2-\lambda \end{pmatrix} = 0$$

Calculating Determinant matrix for the above (it's not a matrix), we get the characteristic equation

$(3 - \lambda)(2 - \lambda) - 2 = 0$

Solving above equation we get the eigenvalues as $\lambda 1 = 4$ and $\lambda 2 = 1$

Substitute eigenvalue $\lambda 1 = 4$ in above equation we get

$$\begin{pmatrix} -1 & 1 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} v1 \\ v2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Thus we get the first eigenvector is

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

For the corresponding eigenvalue λ1 = 4. [39]

The steps followed for image identification are as the follows:

- Initially an adequate amount of face images are obtained. These images are used to train the system for face recognition.
- Using these training set of images Eigenfaces were calculated. It consists of M Eigenfaces with corresponding M largest values are retained. The Eigenface used to create the face-space consist of training set.
- Finally the submitted image is calculated for its Eigenface-weight. Then it's projected to the face space. And the nearest face space is found to confirm the result.[40]

Review of PCA

The image data used for processing is of discriminating signal and with several other images. They are considered as highly noisy because they differ with each other with lighting, facial expression etc. But these face images are presented with some objects like eye, nose, mouth etc. these characteristic features can be extracted by the means of mathematical tool called Principal Component analysis [41].

The faces are represented as a low dimension form in an eigenface approach is derived by applying PCA. The system functions by the projection of face image in space that consists of known images. This significant feature is called as eigenface [42]. The objective of PCA is to take all the variation of the training set of images, and to represent them with some small amount of variables.

The Principal Component Analysis follows a method in matrix known to be covariance matrix (given by m number of rows and n number of columns). Covariance matrix is obtained by mean centre with data across the origin and taking the dot product.

A correlation matrix is obtained by normalising covariance matrix. Normalisation is required since the data in covariance matrix is distributed in a wide range. By doing normalisation it would be easier to assess various component of Principal Analysis.

To understand normalisation, a simple diagrammatic representation is shown:

The diagram represents the number of student's n, the two subject and the marks obtained in the subject1. The first diagram shows the data obtained using covariance matrix. And in the second diagram represents normalisation in which the most of the principal components are obtained just by rotating the axis. By this we can get all the useful values to its maximum [43].

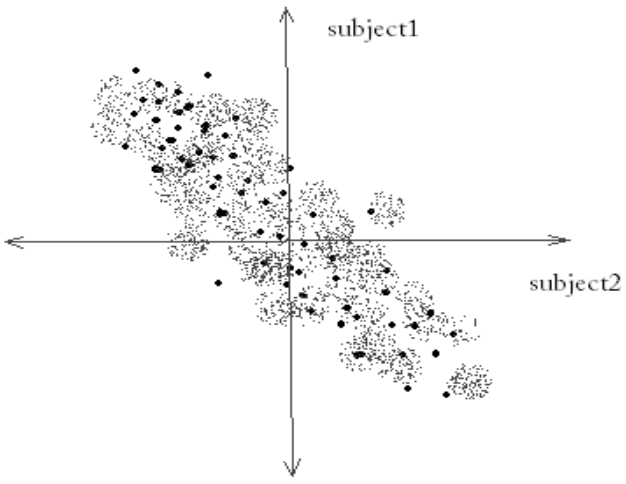


Figure: Centre mark of students in subject1 (using covariance)

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

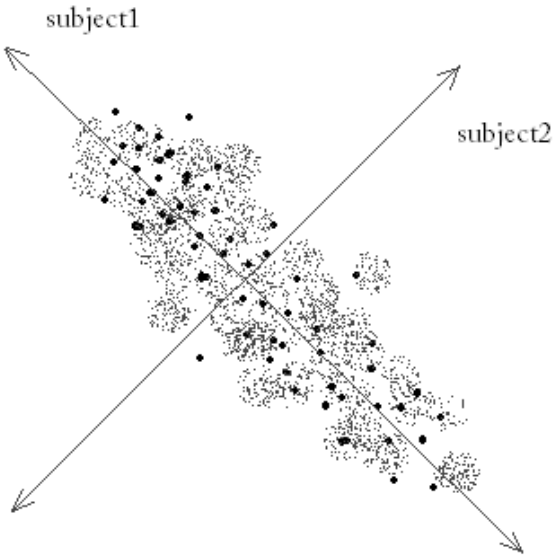


Figure: To get maximum principal component (using correlation)

Explanation of the Theorem Involved

Collection of data:

This is the initial step in which the input for the process is been prepared. The input data is face images. They have been collected with an adequate amount so that it can be used for recognition process.

It's represented in (Γi). Where i = 1, 2, 3…, M. (M – representing images)

Average Face:

The average face Ψ is calculated from the average of all face images

$$\Psi = 1/M \sum_{i=1}^{M} \Gamma i$$

New Image Space:

Every image is different from average face on their distribution, and this is calculated to obtain the distance is by subtracting the average face from all face images. Therefore new image space obtained is

Φi = Γi – Ψ (i = 1, 2, 3… M)

Calculating Covariance Matrix:

The new image space obtained for M Φi, with each of them N^2 * 1 dimension, so that its N^2 pixels. To obtain the Eigenface for the training set of images, we apply the principal component method. Then the covariance matrix is given mathematically by

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T$$

Up to this step we have followed the PCA method of face recognition [44].

To find the eigenvalues from the covariance matrix there are two major disadvantages:

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

- Primary disadvantage is that the matrix itself is very large in size (N × N dimension matrix gives result to N^2 eigenvalues and N^2 eigenvectors.)
- Secondly the computational effort is very big.

Modification of Algorithm:
The simple solution for the above disadvantage of PCA method is that we are reducing the dimension for the matrix. This can be done by calculating M eigenvalues. This scheme was proposed by Turk and Pentland. For example 256 × 256 would result in 65,536 × 65,536 and to calculate 65,536 eigenfaces [45]. We compute a matrix, which consist of M eigenvalues and M eigenvector. So we get a new covariance matrix given as follows.

$$C = \frac{1}{M}\sum_{n-1}^{M} \Phi_n \Phi_n^T = AA^T$$
$$L = A^T A \quad L_{n,m} = \Phi_m^T \Phi_n$$
$$u_l = \sum_{k-1}^{M} v_{lk}\Phi_k \quad l = 1, \ldots, M$$

From the above matrix where L = M × M matrix. υ are M eigenvectors and u are eigenfaces. C is calculated with the equation shown above. The eigenvector calculated represents the variation in the faces [46]

The advantage of this method as explained above, we are calculating only M numbers and not N^2 (since M << N). There is also one more advantage, is that eigenfaces are ranked according to their usefulness. This is been usually done by taking the largest value of eigenvalues (the subset of M is been given by M' eigenfaces).

Classification of face-images:
Identifying faces using eigenfaces can be explained as the eigenvectors which represents one of the dimensions of the face space. The eigenfaces are a group of eigenvectors consists of all the important characteristic of a face. The eigenvectors consist of larger value gives more information than the smaller eigenvalues [47].

When a new image is given as an input say

Γ_{new}

The new image is transformed to eigenface component as shown

$$\omega_k = u_k^T (\Gamma_{\text{new}} - \Psi) \quad k = 1 \ldots M'$$

Calculating the above given we get the new weighted vector as given

$$\Omega_{\text{new}}^T = \begin{bmatrix} \omega_1 & \omega_2 & \ldots & \omega_{M'} \end{bmatrix}$$

Classification done by comparing the feature vector to the library member feature vectored to that of the input image. This comparison is based on Euclidean distance the members and it should be smaller than that of the user defined threshold value Φ_k. The equation for the threshold value defined by the user is given below

$$\frac{\left\|\Phi - \Phi_f\right\|}{\left\|\Phi_f\right\|} \leq \phi_k$$

Where

$$\Phi = \Gamma - \Psi$$

and

$$\Phi_f = \sum_{i=1}^{M'} \omega_i u_i$$

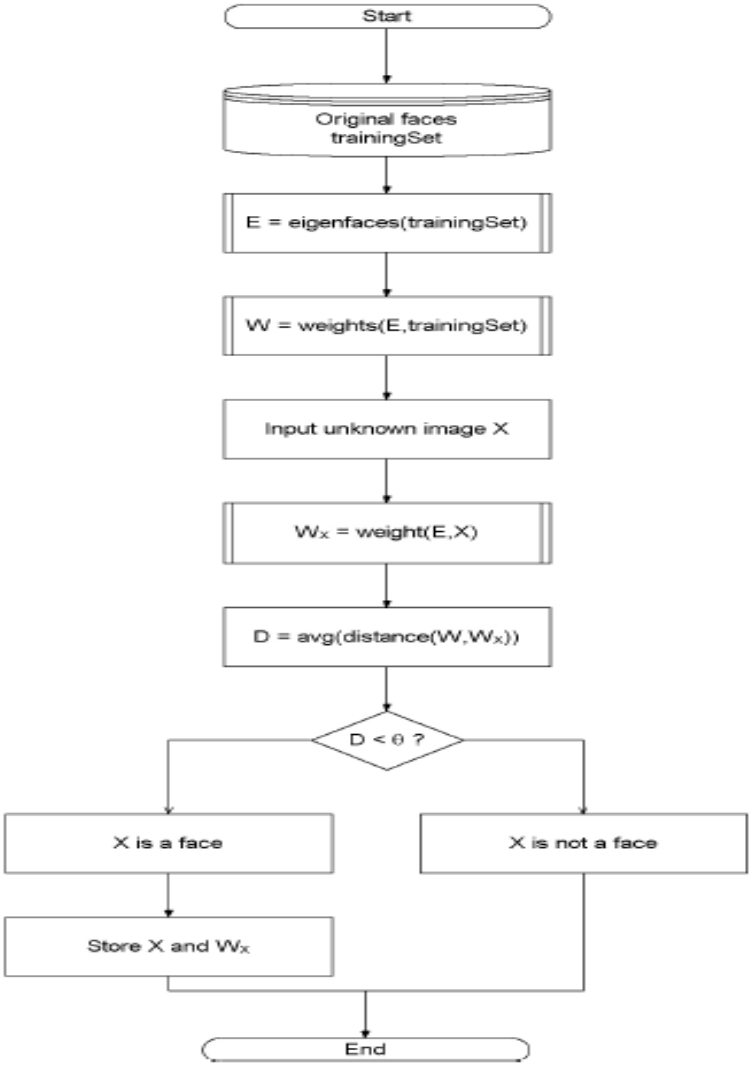
ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

And ϕ_k is the user defined threshold for the input images regarding to k[th] face class. [64]

NOTATIONS

I	Face images
N * N	Size of I
Γ	Training set of images
Γi	i number of training set of images
Γnew	Unknown image, give to find the match
Ψ	Average face
M	Number of Eigenfaces
M'	Number of eigenfaces used for face recognition
C	Covariance matrix
X^T	Transposed X, if X is matrix
U	Eigenvector
Λ	Eigenvalue
W	Weight
Ø	Threshold value
Ωi	Weighted vector for the image i

Base Algorithm



Base algorithm for eigenface recognition [48]

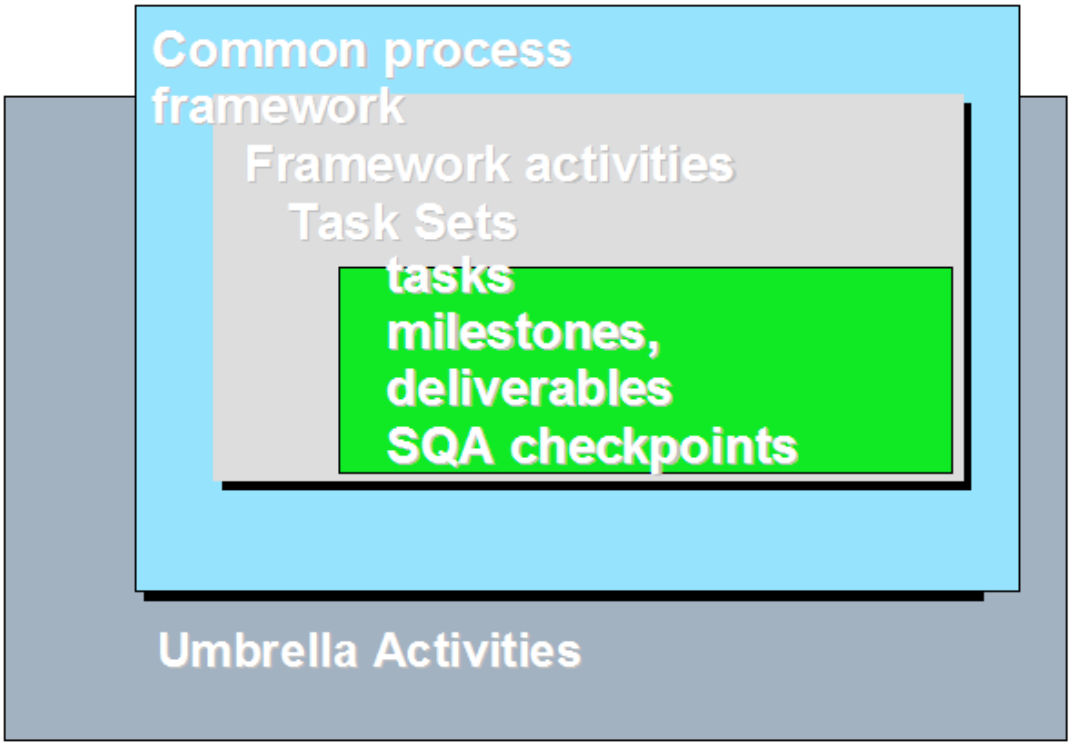
ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

PROJECT PLANNING

Proposed Methodology and Software Life Cycle for the Project

Software project plays very important role in our life and business. Almost each and every business uses software to enhance there performance. So generally a "relationship must exists between the quality of product and quality of a process"[SDLC]. Software process development is not to only to produce to good quality software but it refines the system through which software is made.

"Software methodology is a process framework which is established by defining a small number of framework activities which is applicable to all software projects"[SE]. Each small framework gives us different project deliverables. A software methodology is choose on the basis of the nature of the project, the methods and tools which we are going to used and the control and deliverables which are required .



There are lots of software methodology which are available these are, Linear sequential model, prototyping, incremental model, Spiral Model etc. I have created this model using Linear sequential model.

Why Linear Sequential Model

Most of the software project are being created using this method. I used this model because it is a "sequential approach to our project which will start from system level and progresses through analysis, design, coding, testing and maintenance"[SE]. I choose this model because each phase of linear sequential model has number of goals and I can't move to next phase until I completed the goal of each phase. Each phase linked with an arrow that shows the flow of information between each phases. I used this because model because this model is best for software project if we have 3 – 6 months time. If I found errors in the phase in to order to resolve that error I have to move upward. Suppose If I found my any error in coding phase I have to move to design phase to lookup if I didn't find any error there then I have to move upward to analysis phase. The major drawback of this model is that we can't see the initial picture of our project. This model is also called waterfall model.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

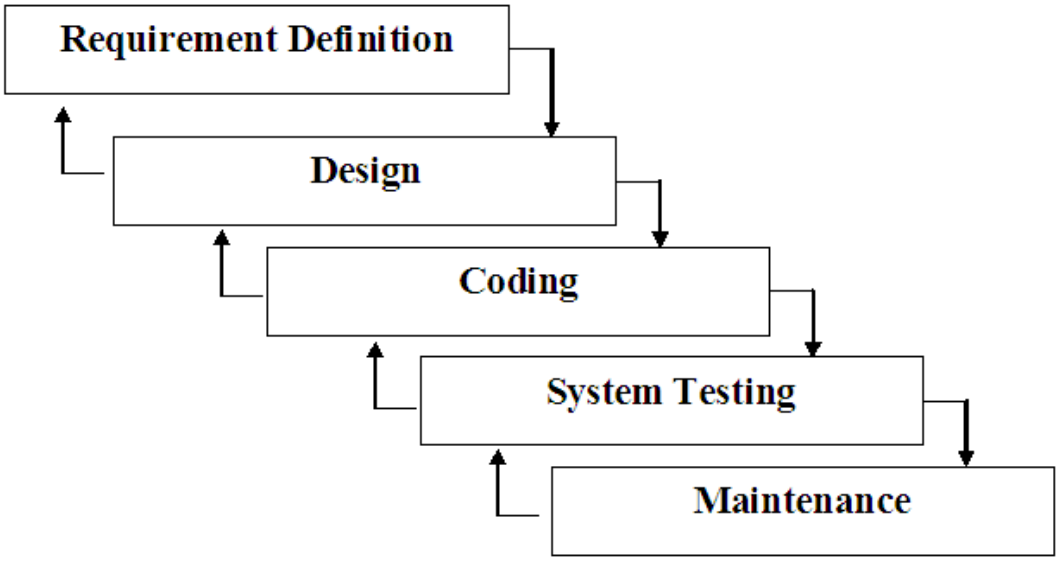


Figure: Linear Sequential Model

Software requirements analysis
This is most time consuming phase of this model. This phase is to gathering information and focused on software. To understand the nature of programs to be built we must know the full requirement of system. In this phase we gather lots of information from several sources. In this phase we perform system planning and make feasibility report. This phase also tells that by which methods and software tools we will use to make this project.

Design
Software design is actually a multi step process. The design process translates requirements into representation of the software. In design phase we create system context level diagram, behavioural model which tells us that who are interacting with the system from outside and inside. In this also define our use cases and state transition diagram through which we know the flow of our software project. In design I also define algorithms for my project then I will convert those algorithms into my code.

Code generation
In code generation phase we develop our software from the knowledge we collected from analysis and design phase. If design is performed in a detailed manner, then coding can be accomplished automatically using IBM rational rose. If we defined our class diagram perfect and sequential diagram and use case model in perfect manner then IBM rational rose generate code for our project. This can be change according to our need which will save lot of time.

Testing
Once the code has been generated the testing begins. "The testing process focuses on the logical internals of the software, ensuring that all statements have been tested"[SE]. In this testing we do testing using two testing method. These are white box and black box testing. If we found any error then we remove those errors and move to next phase.

Maintenance
There is a possibility that there may be an error or change is required after we dilever our project. So the software must be adapted to the external environment, the changes that may occur or if the customer wants to enhance the functionality of the system. Software support/maintenance reapplies to each of the preceding phases to an existing program.

Design using SELECT Yourdon
In this part the overall real time system analysis and design will be discussed. It makes a clear understanding of how the systems are built and its all functions.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Context Diagram:
Context diagram can be explained as, which gives the base model for the system development. The context diagram for the eigenface recognition system is shown as below:

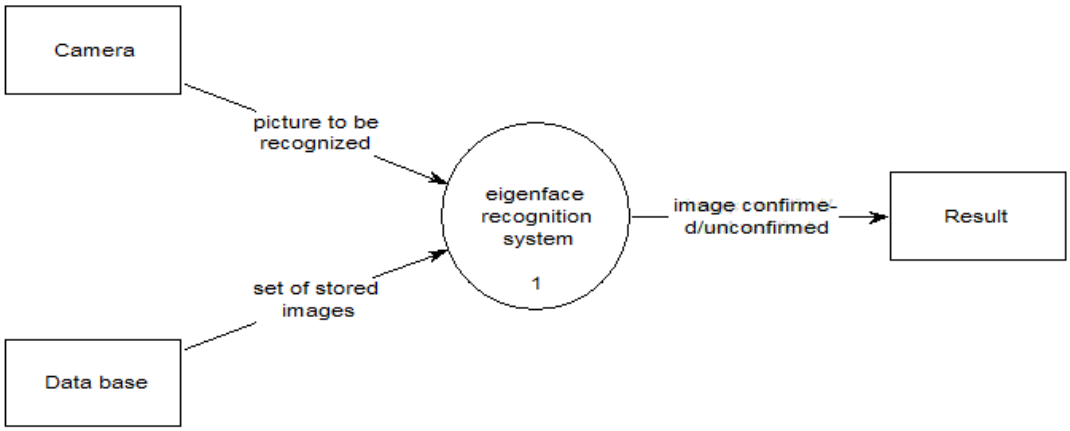


Figure: Context Diagram

From the above diagram, we can clearly observe the basic requirement for a face recognition system. The main notable things are the external events that take place. In this face recognition system which uses the input from the database and the camera. And the system performs the logical functions and gives the result.

Behavioural Model:
The Eigen face recognition based on Principal Component Analysis is shown in the behavioural model diagram. This model summarises the whole process of face recognition using Eigen faces.

Initially all images given to the system are checked for the given specific criteria (all should be in same size). The eigenspace is calculated using the set of training images. Using these set of images the average face is also calculated. Difference of number of images and average face gives the covariance matrix. These eigenfaces obtained in these processes represents the various features of the face.

Normalising eigenvector or sorting the eigenvectors according to the highest variations. This makes the dimension to become low than the original image. After calculating the eigenvalues for both the input and saved images, the two sets of images are projected to the face space.

The distance between weighted vectors is calculated using Euclidean distance. The value obtained is compared with that of the threshold value, thus the matching of image and its result is given. Finally, depending upon the False Acceptation Rate and the False Rejection Rate the threshold value is changed [49].

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

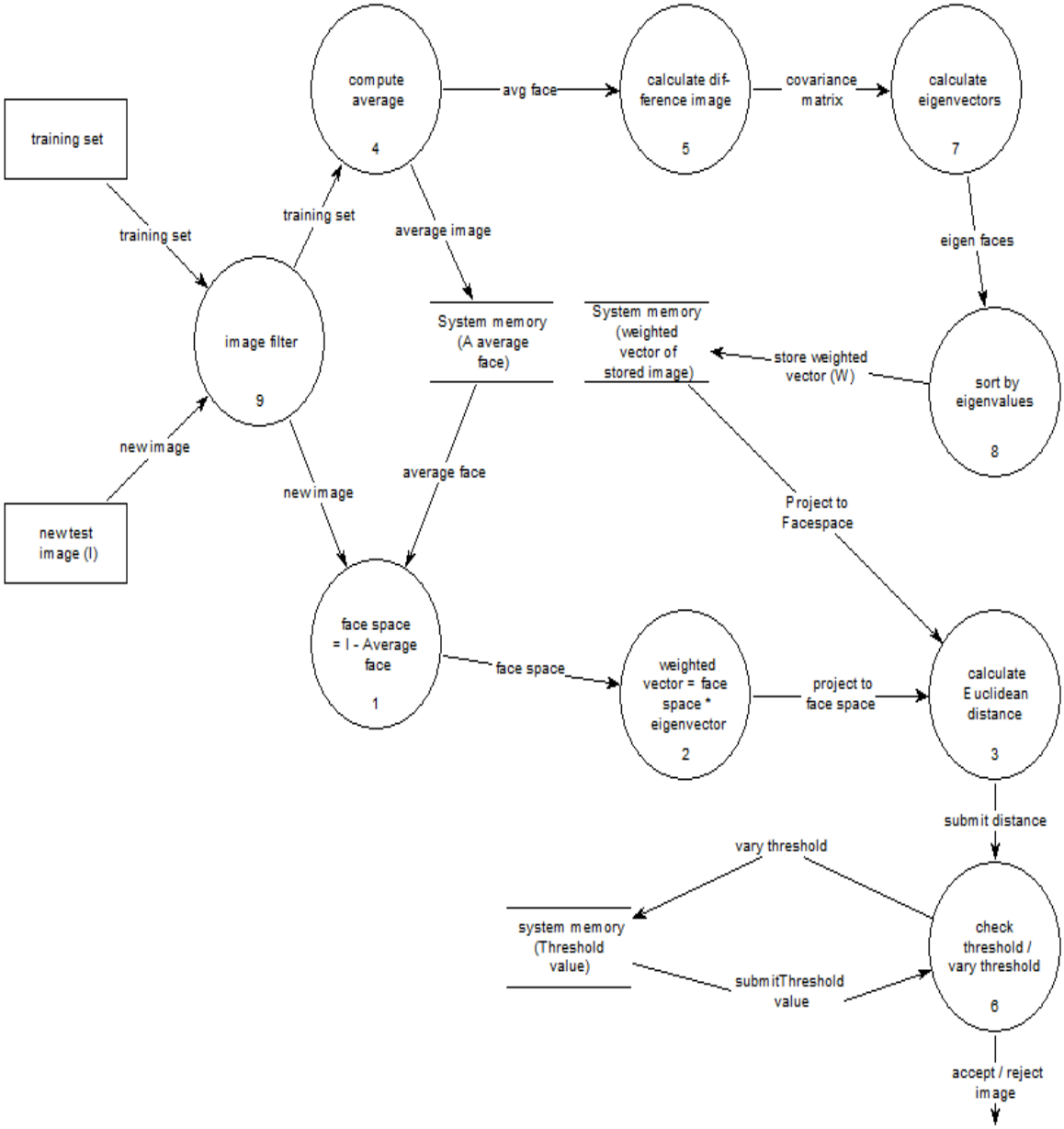


Figure: Behavioural model for Eigenface Recognition System

User Interface Diagram

The UML (Unified Modelling Language) is used to create the visual model for this software project. User interface diagram, class diagram and the sequential diagram represented in this paper is created using the software designing tool IBM Rational Rose, version 7.

UML can explain as to create a visualisation of a design and checking against all the requirements before stating of coding [50]. In detail, it's not only a representation of diagrams, but it's a language for capturing and expressing knowledge about the subject for the purpose of communication. It is also unifying the information systems and technology industry's best engineering practise [51].

In this project considering the user interaction diagram of UML design. The user is assumed as the admin and the user case given is to compare faces. In comparing faces the user is allowed to give the path for the stored training images and the input image to be matched. And the system gives the output to the user, either it finds a match or not in respective to input image.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Figure: User case diagram

Class Diagram

The static structure of a system is explained by class diagram or the structure of the system as a whole rather than how it behaves.
The class diagram consists of the following:

Classes: classes represent the features of Entities. The features included are associations, operations and attributes.

Associations: the relationship between two or more classes is represented by associates and they have common characteristic.

The class diagram for this project is represented as following:

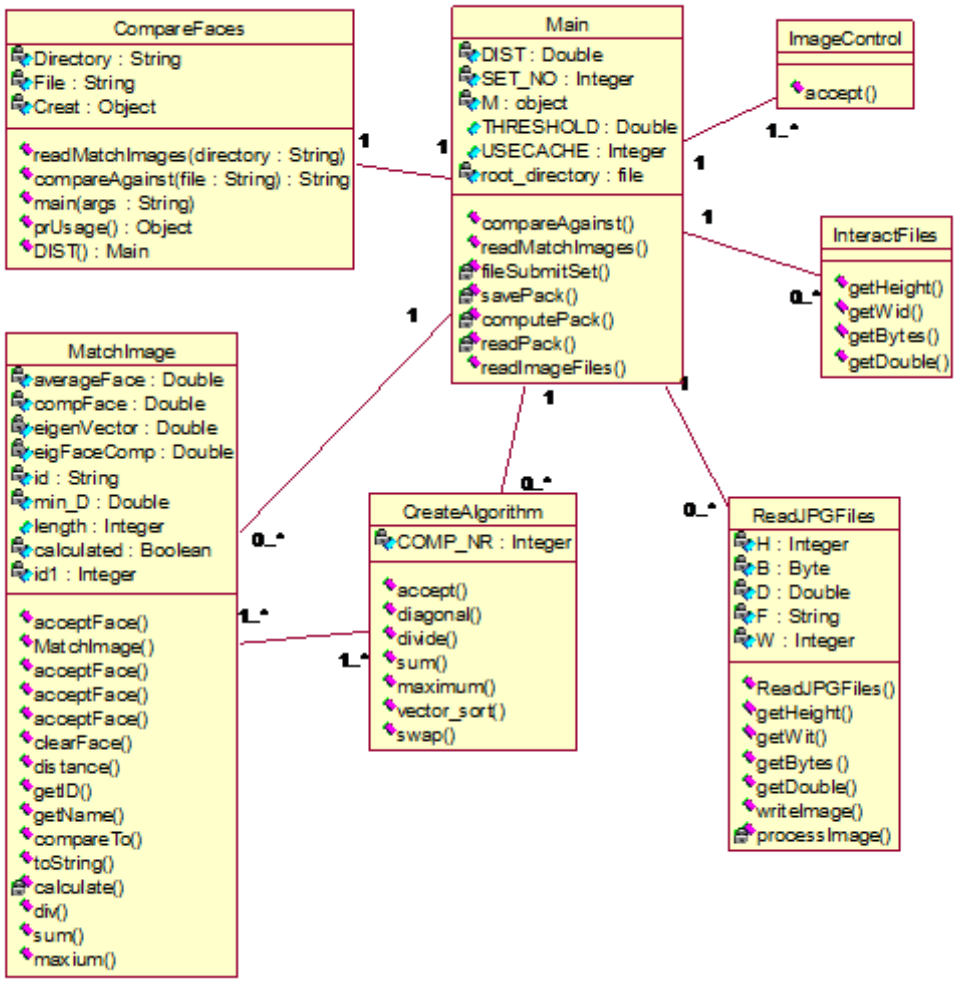


Figure: Class Diagram

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Sequential diagram

The sequential diagram explains the interaction between the classes. The sequential diagram for this project is shown as following

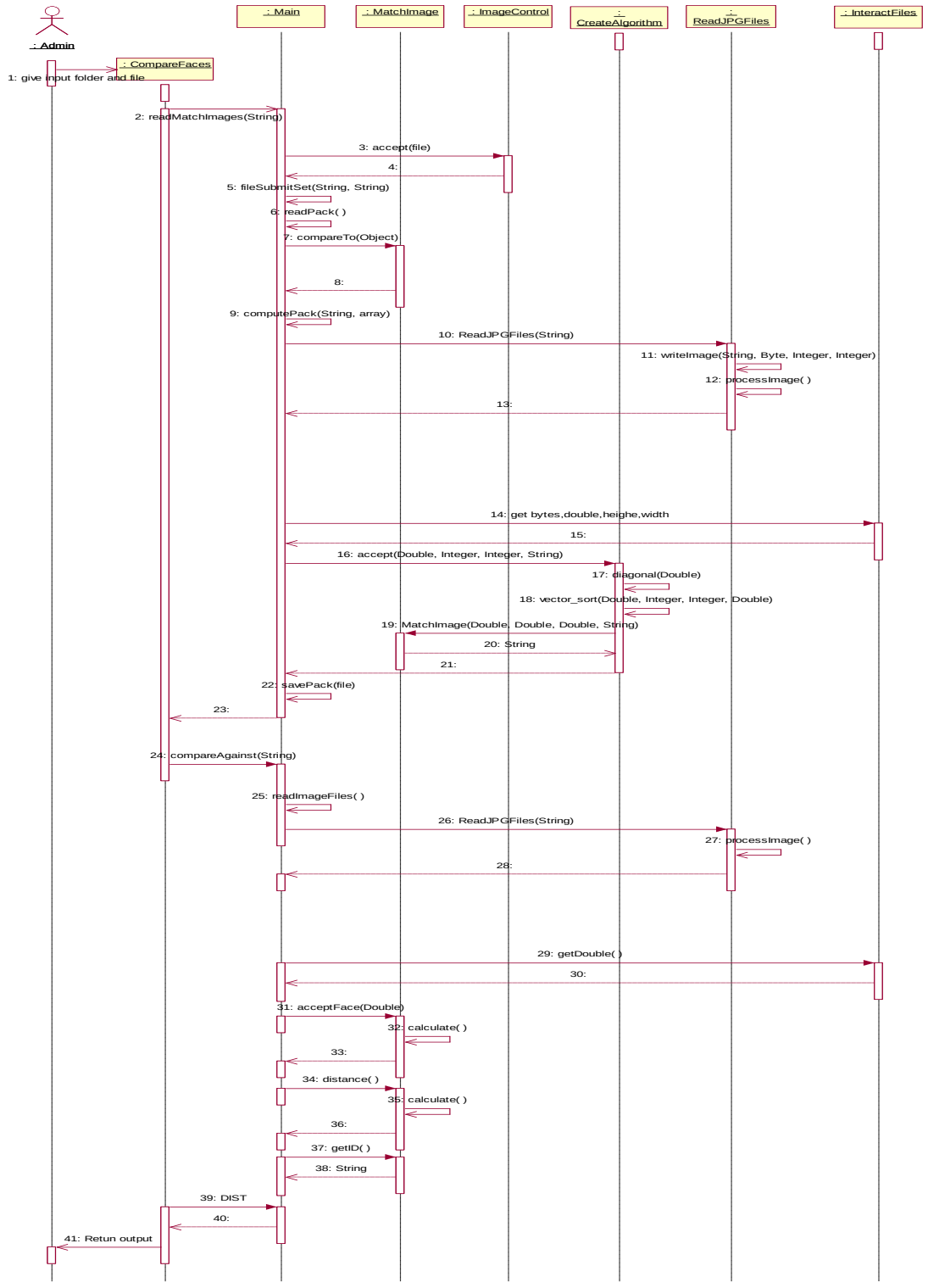


Figure: Sequential Diagram

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

State Transition Diagram

State Transition diagram explains the behaviour of the system to external stimuli. The state chart for this face recognition is shown as following.

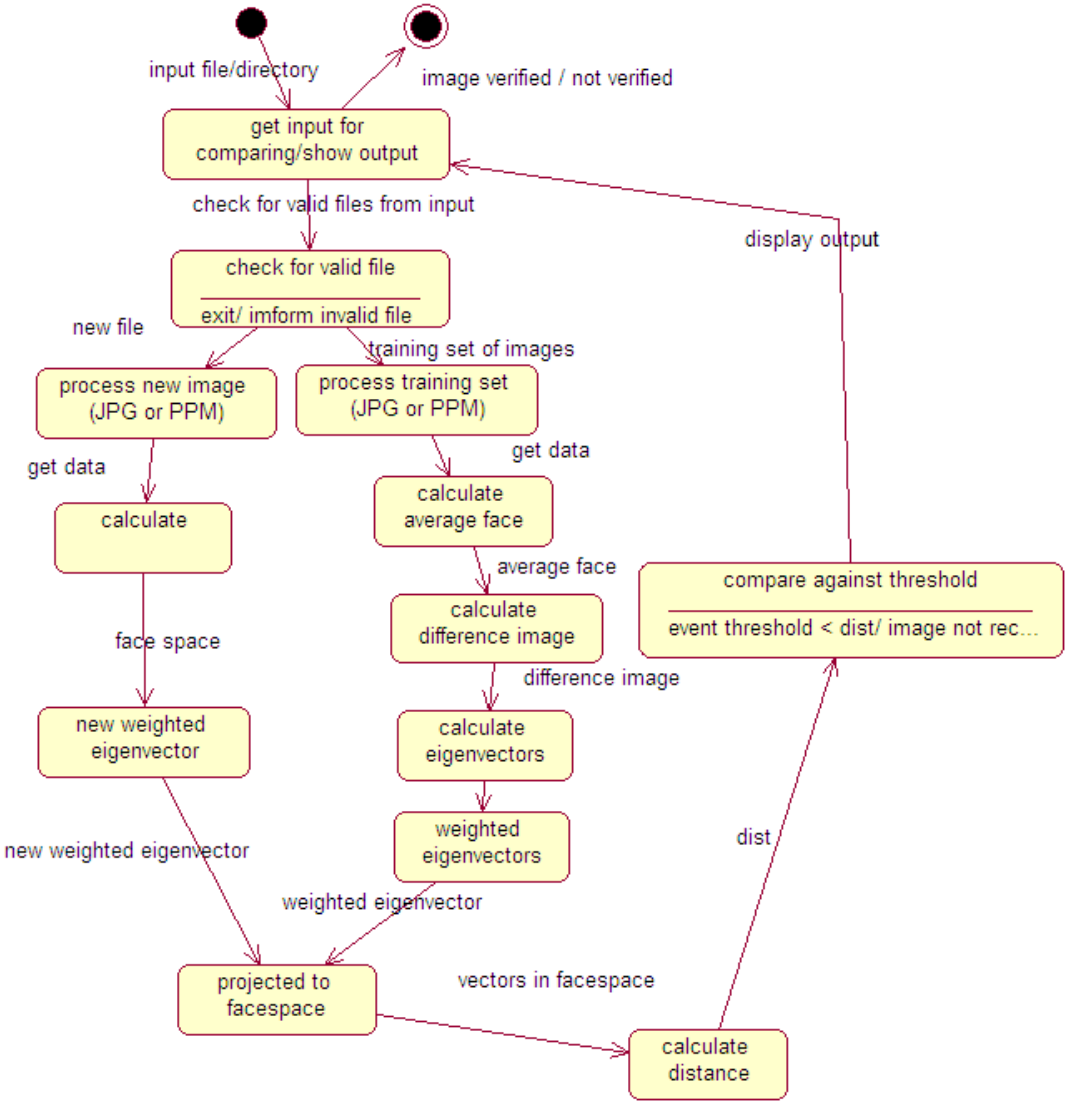


Figure: State Transition Diagram

PROJECT EXECUTION

Writing Program and Explanation

After the successive Analysis and Design for the system. Implementation of the designed system to a real-time function is done by programming. By following the above design the face recognition system package consist of major objects which are been explained as follows.

Class Compare face:

The object or the class **CompareFaces.java** is the main class which interfaces with the user and also call the other objects for the process of face recognition. The input from the user is taken as two arguments. First argument is the path for the database (in this system it uses a directory as a database) and second argument is the input file given by the user for identification. This information provided from the user is passed to the object **Main.java**, where it is related to all other classes as for image processing and matching images using Eigenfaces.

Finally it receives the result and the result would tell the user either the given file has a match or not.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Class Main:

Functions for Training set of images: The path for the training set of images is obtained from the user entry. Initially, all the files are listed and sorted in order.

Then these files separated with into groups, so that, it would be easier when it contains bulky number of files (usually gives a memory error).

Set of images submitted are now checked with the cache file in the same directory. If it finds the relevant cache files with respect to the submitted set of images, these files are read to find the objects in them and submitted to match against the submitted file (matching performed in **MatchImages.java**).

When there are no cache files or no appropriate cache files are found, the set of images known as pack will be first to be sent to find that all files are having the exact file extension (i.e. all are JPEG file types) and also checked that all images are of the same size (height and width should be the same). Then the image is been send to the class **ReadJPGImages.java** to process the image files where the image data is retrieved from interface class **InteractFiles.java**.

Data obtained double, bytes, height, and width form overall image processing forms the basis for Eigenface calculations. Thereafter the image data are sent to calculate the Eigenfaces (done in the class **CreateAlgorithm.java**). Later on, it saves the calculated values catches from the cache while the matching images are done in class **MatchImages.java**.

Functions for input new image:

New image or the image that is to be harmonized against the images stored is obtained from the user input. This single image is sent for the image dispensation, where it is primarily checked for the file type. Here the file is processed using **ReadJPGFiles.java** and the double is obtained from the interface **InteractFiles.java.**

The data obtained from the image file is double array of the face. This data is sent to class **MatchImages.java**, where it calculates Eigenface for the input new image and compares with that of the Eigenfaces for training set of images, and returns the smallest distance found.

Finally, the returned distance is compared with the threshold value, so that, determines the input image has a match or not.

Class Image Control:

Class **ImageControl.java** is used as a filter to check the specific directory that contains valid image files, which are used for training the system. It works simple as image files passed from the **Main.java.** These images extension are obtained and filtered for just checking their extension. Thus, only images having the extension mentioned is passed for further processing of image in class **ReadJPGFiles.java** or the program terminates saying it has varied files.

Class Read JPG Files:

Image processing is done in this class. When an image is passed to this class, it uses Java native method to encode and decode images (this method may by deprecate or may not available in future version of JDK) [62]. The image is encoded so the to get the RGB values in 8-bit. The mixture of these three bytes gives the single colour. Such that, 255 255 255 - white, 0 0 0 - black, 255 0 0 - red, 0 255 0 - green and 0 0 255 - blue. [52]. Then the image is decoded further to obtain the height and width. And also, byte array which is usually height multiplied with the width, which is in bytes. After all, 8-bit RGB data is converted to 64-bit double. And all of the values height, width, bytes and double are returned to interface **InteractFiles.java**. For more libraries in Java for pixel reading for images refer [53].

Class Create Algorithm:

The calculations of face-space are done in this class **CreateAlgorithm.java**, this face-space obtained is used for the process of face recognition. Face-space is defined by "EIGENFACES". Eigenface represented by number of eigenvectors of the face, where these eigenvectors are the principal component of the face image [54].

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

The eigenvectors are obtained from covariance matrix. To calculate the eigenvalues and their corresponding eigenvectors in Java, a special package is required known JAMA (A Java Matrix Package). This package provides adequate functionality for matrix related operations [55].

The actual image matching is made in the class **MatchImages.java**, but this process needs a lot of calculations such as average face, covariance matrix, eigenvectors, eigenvalues and the largest eigenvalues and its corresponding eigenvalues. And also, the weighted vector for the input new image.

The input accepted are the face array consist 2-D array with sixteen rows and columns of all the same size, the width, height and the ID of the images. Following the Eigenfaces theorem explained above. First step is to calculate the average face. The sum of training images (given in array), is divided by the total number of training set gives the average face.

A new image space is now obtained by subtracting the average face from the set of training images which is in the form of arrays (M × N^2). And this image space is array copied for further use to get the weighted vectors (principal component vectors wk).

Covariance matrix is formed by calling the matrix function transpose(), to calculate covariance we have to determine face-array, length of the face-array and length. Therefore we get the covariance matrix (M × M). For this covariance matrix eigenvalues and eigenvectors are obtained by using the class Eigen value decomposition from JAMA package [63].

All the eigenvalues obtained are enumerated and sorted down, such that, we are getting only the highest values of eigenvalues and their corresponding eigenvectors.

From the above step we obtain the eigenvector for covariance matrix (M × M), this eigenvector is multiplied with that of the image face obtained (M × N^2). By doing this we get Eigenfaces for (M × N^2). The Eigenfaces obtained are normalised by dividing each of them with the maximum value of eigenvector obtained.

Finally, the weighted eigenvectors of a face is calculated. It is done by manipulating number of training set of images to the specific number we mention that would be sufficient for face recognition. Therefore we get the weighted number of Eigenfaces or face-space. The obtained values face-space, average face, eigenvector and ID are returned to the class **MatchImages.java**, for further matching of images.

Class Match Images:
The face-space submitted by class **CreatAlgorithm.java** contains all the information required for face recognition. Matching images in this class MatchImages.java is quit faster. Each of the face-space pack consists of sixteen images and their identifying information. We also made this process serialized so the further computation using **CreateAlgorithm.java** is neglected.

The face-space pack submitted consists of average face array, Eigenfaces of images, eigenvectors and the names for identifying every individual files. The face array is used to normalise the face image to be matched against. The eigenface components are used to project to the eigenspace. Eigenvector onto the matched images will be projected to. An array of strings which holds the identification for the set of images represented in the face-space.

The input image obtained, after image processing form class **Main.java**. This, input image data is converted an array of double, and passed to the calculation part. During calculation the image are projected to the face-space. So that, to fine the minimum distance of matching image.
The normalised face array is subtracted with average face, the result gives the face-space for the input image. The face-space is then multiplied with the eigenvector to obtain the weighted eigenvectors of the input face.

To identify the image, we find the minimum distance between the vectors. This is done by **Euclidean Distance** [56]. To explain the calculation in more detail about Euclidean distance that's to deal with vector, a small example is given:

System.arraycopy (dist,0,min_Distance,0,COMP_NR);

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Euclidean distance means the metric, of distance between the same lengths of two vectors. The above array copies form the 'dist' to 'min_Distance' and the length is COMP_NR.

Distance between two vectors "input image" and "training images", the resulting "dist" is also a vector. So, each component of the distance vector is equal or absolute value to the corresponding operands.

{4,4,3,2} .distanceFrom({1,2,2,3}) = = {3,2,1,1}

The above is just as dist = input images – training images. So we repeat the calculation with the group of training set of images to get the minimum distance. Obtained minimum distance is returned to that of the **Main.java**. Thus, the operations performed in matching images.

JAMA:
A Java Matrix Package is used for the programming. This is because JAMA provides numerical linear algebra operations for the fundamental of matrix operations. It dose number of operations that involved in real matrix.

About copyright: "This software is a cooperative product of The MathWorks and the National Institute of Standards and Technology (NIST) which has been released to the public domain. Neither The MathWorks nor NIST assumes any responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic" [55].

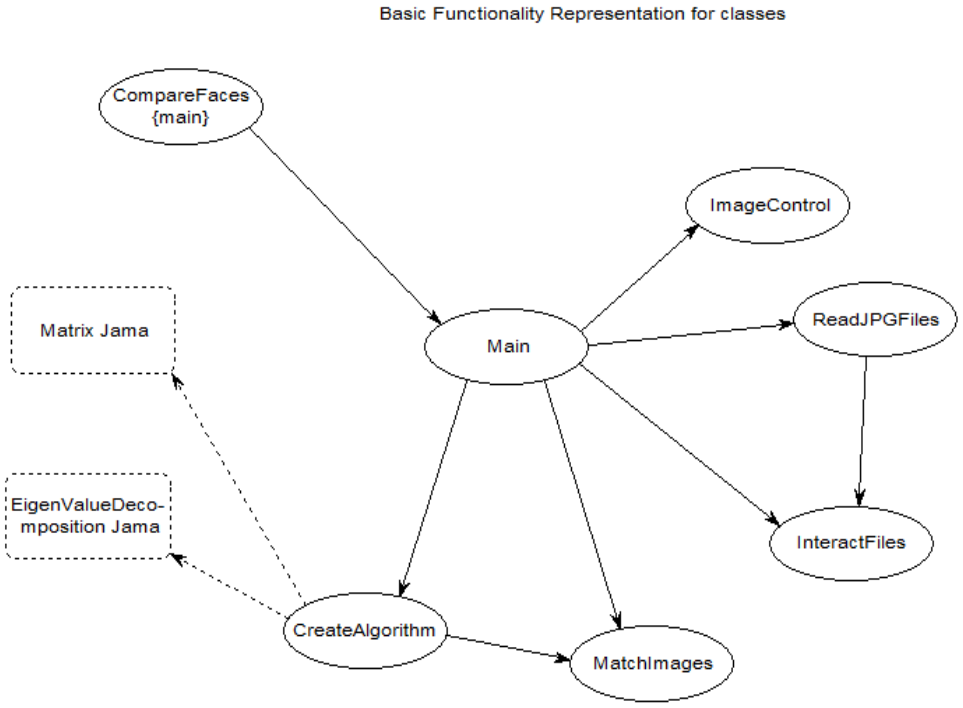


Figure: class's interaction diagram

Implementation
The implementation process for face recognition in this project which requires the following

- Java SDK (Software Development Kid) :
  The overall programming involved using Java programming language. The program is done using NetBeans IDE (It's a free open source Integrated Development Environment used for software development. It provides tools to create professional desktop, enterprise, mobile application with Java, c/c++ etc., and many more [57]

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

SDK are required to create an application for software package, they depend upon the hardware, computing system, operating system etc., Java SDK is required as an application program interface. It interfaces with the program to provide a run-time environment so as to perform face recognition.

Java SKD is a product of Sun Microsystems; it's been licence under GNU General Public Licence (GPL). It can be downloaded according to the operating system requirement for its different applications [58].

- JAMA (Java Matrix Package):
  Java Matrix Package as explained earlier used to perform matrix operations with Java. This package is available from http://math.nist.gov/javanumerics/jama/ and the library files to be installed in the bin folder of Java SDK.

  We also create a class path to set an argument in a command line. This is done, so that the Java virtual machine find the user defined classes or the user define package.

  Java Matrix Package is given in the class path of the system. This is done because it's not a default library function within standard Java. So, when we run the program the Java finds the user defined class path and gets the all the library function within them to run the program. There for the program can be called any where from the computer.

  The class path is given from a computer by the following steps
  My computer – properties – advance- environment variables – path and finally edit the path which we want to include in the class path.
  i.e, C:\Program Files\Java\jdk1.6.0_07\bin\jama.jar is included in the variable in the above selected path [59)].
- Training Set of Images:
  For the process of face recognition it requires a set of images it be stored. These images are used to compare against the input image given by the user.

  This project is implemented in HP notebook. So, HP web cam is used to obtain the picture of an individual. The HP webcam is 1.3 mega pixels and used inbuilt software QuickPlay to operate [60]. Using this, different snaps of an individual are obtained. Therefore, images for training set stored in database and images for comparing against training set is obtained.

TESTING AND RESULT

Software testing is necessary for a project. Testing is made so as to evaluate and get the result for developed software. Software testing is made is controlled form with all normalities and abnormalities deals with the system.

In this project, concerned with, face recognition system. Its primary requirement is a database that contains number of face images. This database is used to train the system for face recognition. Secondly, an input image that is to be matched with the images stored in the database.

The input image given to the system is variable according to the user input given. So in the testing made for this project

S.No	Test case	Expected Result	Actual Result	Pass/Fail
1.	Image to be compared is sampled of the same person at a similar distance and lighting conditions.	Input image is compared with the sample images and the correct image is matched.	Same as expected	Pass
2.	No image of the person compared exists in the sample images	The application should report no matches	It exits saying "No images Found"	Pass

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

3.	Two copies of the same image exist inside the folder.	Comparison should exit with the first matched image of one of the three copies. The distance should be reported as 0.	Same as expected	Pass
4.	Images of the same person with different facial expressions are compared. One with a smile and another grim.	The matching distance should be reported as appropriate to the threshold being set. If the threshold has been set as high as 10, the images to be matched with success and otherwise no match should be made.	The matching depends on the amount of variation in the expressions and the threshold value set.	Pass
5.	The images are sampled under different lighting conditions one with bright lighting and another dark.	The images should be compared as according to the threshold set.	Same as expected.	Pass
6.	Images being compared are sampled at different distance.	The comparison should not identify the other image as a match	Application says "No match found"	Pass
7.	Same person's image is captured with different backgrounds i.e. plain and patterned background.	The faces should be matched at reasonably higher thresholds.	This is non-deterministic. The faces are matched with a huge distance or not matched at all.	Fail
8.	Images of the same person with and without cap	Images should not be matched	Same as expected	Pass
9.	A copy of an image is resized to a different dimension and tested.	Images should not be matched being different dimensions.	Same as expected	Pass
10.	Images of the same person captured at different angles are compared	Images should not match as there would be variance in the calculated Eigen faces	Same as expected	Pass

IMPROVEMENT AND EXTENSION

Improvement and future work is done considering the more efficient result in face recognition, and also, considering creating user friendly environment.

Automated system: The project can be further extended to form an automated system in which a cam is connected to the system which is active all the time. This webcam is controlled by a separate program which looks for any movement for an object. When it identifies any movement it captures a sequence of images and face is identified from these images. The face obtained is then used for face recognition.

Controlling a camera is done by using JMF (Java Media Framework); JMF is an Application Program Interface which enables audio, video and media interface that can be used for Java application and applets [61].

The future work also includes creating a better user interface. By this user can have a better view graphically, it also enables the user to interact with the system easily. This includes that the user can view the output result of how Eigenface output are given by representing them in a graphical output. This graphical method makes simple so that user doesn't have to use the command line mode to run the program.

Selecting the number of Weighted Eigenvectors (Wk) can change the result to a higher level. When the number of Eigenvector is more the input becomes very noisy. When it is very low then the percentage for identifying an image becomes low. So it is necessary to make number of experiments to set an exact value for the weighted Eigenvectors used for image matching. The future improvement of this project also includes adding one or more

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

algorithm to perform face recognition. It means two separate algorithms are used to provide to do the same face recognition process.

DISCUSSION

The project based on face recognition is selected, because the research involved in this topic, which covers all fields in a software development. And also the study made for face recognition is challenging and more interesting all the while.

This project based upon face recognition which has a multiple tasks to be performed. So, in general identifying a face using computers seems to be very hard, as it has complex ways to achieve matching images. But, after learning and understanding the basic mathematics involved in this project, which makes everything easy than expected. And also, learning the theorem involved in this project gave a strong basic to do this project.

Before starting to the actual coding of the program, I made an outline structure for the program. There were two software development tools where used for this process. The tools are SELECT Yourdon and IBM Rational Rose enterprise edition. These tools gave a strong base for the software development process. By doing this the entire source are easily identified, as well as we get a total outline for the program to be developed. The designing part also helps to break down the program into other small parts, this give more advantage while programming.

The coding part is done progressively after studying and adopting counter parts of previously created sample programs. These studies, of all these counter parts of the program separately are combined to form all the process of this specific program face recognition. While coding, the critical part is to apply the theorem regarding Eigenface to the programming. But this can be done initial, by simply an executable program that calculates the Eigenface. As the progress the project follows the software methodology used, we will apply every necessary steps required. This makes the effective face recognition system.

CONCLUSION

In this project I have made as effort to successfully complete a working model for face recognition system. I have selecten the method introducd by M.Turk and A.Pentland in 1991, this method popularly known as Eigenface. By setecting this method I was able to study and progress a simple

REFRENCE

[1] Davis, Ellis, and Shepherd (eds.), Perceiving and Remembering Faces, Academic Press, London, 1981
[2] Effects on face images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 442–455, 2002
[3] Multi-lane architecture for eigenface based real-time face recognition.Rajkiran Gottumukkal, Hau T. Ngo, Vijayan K. Asari. Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA 23529, USA
[4] Image Analysis for Face Recognition Xiaoguang Lu Dept. of Computer Science & Engineering Michigan State University, East Lansing, MI, 48824
[5] http://www.users.cs.york.ac.uk/~juquan/A%20Literature%20Review%20of%20Face%20Recognition.ppt#262,11,Nonlinear analysis
[6] J. Winkeler, B.S. Manjunath, S. Chandrasekaran, Subset selection for active object recognition, in: CVPR, vol. 2, IEEE Computer Society Press, Silver Spring, MD, 1999, pp. 511–516.
[7] H. Moon, P.J. Phillips, Computational and Performance aspects of PCA-based Face Recognition Algorithms, Perception, Vol. 30, 2001, pp. 303-321
[8] http://cmp.felk.cvut.cz/cmp/software/stprtool/examples/pca_example.gif
[9] M.S. Bartlett, J.R. Movellan and T.J. Sejnowski, Face recognition by independent component analysis, IEEE Trans. Neural Networks 13 (6) (2002), pp. 1450–1464.
[10] Sequential row–column independent component analysis for face recognitionNeurocomputing,by (Quanxue Gao, Lei Zhang, David Zhang)
[11] 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, AVBPA'99, 22-24 March 1999, Washington D.C., USA, pp. 211-216
[12] P. Comon. Independent component analysis, a new concept? Signal Processing, 36:287–314, 1994.
[13] Face recognition using LDA mixture model Pattern Recognition Letters, Volume 24, Issue 15, November 2003, Pages 2815-2821 Hyun-Chul Kim, Daijin Kim, Sung Yang Bang

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

[14] S.J. Raudys and A.K. Jain, Small sample size effects in statistical pattern recognition: recommendations for practitioners, IEEE Trans. Pattern Anal. Mach. Intell. 13 (1991), pp. 252–264.
[15] Improved discriminate analysis for high-dimensional data and its application to face recognitionPattern Recognition, Volume 40, Issue 5, May 2007, Pages 1570-1578 Xiao-Sheng Zhuang, Dao-Qing Dai
[16] http://www.cc.gatech.edu/~dminn/mini-proj.html
[17] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection, Proc. of the 4th European Conference on Computer Vision, ECCV'96, 15-18 April 1996, Cambridge, UK, pp. 45-58
[18] Afast kernel-based nonlinear discriminant analysis for multi-class problems, Yong Xua,b, David Zhangc,∗, Zhong Jinb, Miao Lia, Jing-YuYangb,Pattern Recognition, Volume 39, Issue 6, June 2006, Pages 1026-1033
[19] http://www.kernel-methods.net/tutorials/KMtalk.pdf
[20] http://www.kernel-methods.net/tutorials/KMtalk.pdf
[21] http://www.snl.salk.edu/~fellous/posters/Bu97poster/BUPoster.pdf
[22] http://www.face-rec.org/algorithms/EBGM/WisFelKrue99-FaceRecognition-JainBook.pdf
[23] http://www.cs.colostate.edu/evalfacerec/papers/EBGMThesis_Final.pdf
[24] View-based active appearance models, T. F. Cootes, G. V. Wheeler, K. N. Walker, C. J. Taylor, Volume 20, Issues 9-10, 1 August 2002, Pages 657-664
[25] Zhao et al., 2003 W. Zhao, R. Chellappa, P.J. Phillips and A. Rosenfeld, Face recognition: A literature survey, ACM Comput. Surveys (CSUR) Archive 35 (4) (2003), pp. 399–458.
[26] Phillips et al., 2007 Phillips, P.J., Scruggs, W.T., O'Toole, A.J., Flynn, P.J., Bowyer, K.W., Schott, C.L., Sharpe, M., 2007. Report of FRVT 2006 and ICE 2006 Large-Scale Results, Tech. Rep. NISTIR 7408.
[27] V. Blanz and T. Vetter. A morphable model for synthesis of 3D faces. In Computer Graphics Proceedings SIGGRAPH, pages 187{194, Los Angeles, 1999.
[28] J. Huang, B. Heisele, V. Blanz, Component-based Face Recognition with 3D Morphable Models, Proc. of the 4th International Conference on Audio- and Video-Based Biometric Person Authentication, AVBPA 2003, 09-11 June 2003, Guildford, UK, pp. 27-34
[29] D. Gorodnichy, Video-Based Framework for Face Recognition in Video, Second Workshop on Face Processing in Video (FPiV'05), Proc. of the Second Canadian Conference on Computer and Robot Vision (CRV'05), 09-11 May 2005, Victoria, British Columbia, Canada, pp. 330-338
[30] S. Zhou, V. Krueger, R. Chellappa, Probabilistic recognition of human faces from video, Computer Vision and Image Understanding, Vol. 91, 2003, pp. 214-245
[31] M. Isard, A. Blake, Contour tracking by stochastic propagation of conditional density, Proc. ECCV (1996) 343–356.
[32] G. Kitagawa, Monte carlo filter and smoother for non-gaussian nonlinear state space models, J. Comput. Graphical Statistics 5 (1996) 1–25.
[33] J.S. Liu, R. Chen, Sequential Monte Carlo for dynamic systems, J. Am. Statist. Assoc. 93 (1998) 1031–1041.
[34] D. Gorodnichy, Video-Based Framework for Face Recognition in Video, Second Workshop on Face Processing in Video (FPiV'05), Proc. of the Second Canadian Conference on Computer and Robot Vision (CRV'05), 09-11 May 2005, Victoria, British Columbia, Canada, pp. 330-338
[35] http://www.glenbrook.k12.il.us/gbssci/Phys/Class/1DKin/U1L1b.html
[36] http://www.sosmath.com/matrix/matrix0/matrix0.html
[37] http://www.sosmath.com/matrix/
[38] http://www.maths.surrey.ac.uk/explore/emmaspages/option1.html
[39] http://www.cs.iastate.edu/~alex/classes/2007_Spring_575X/slides/13_EigenFaces.pdf
[40] http://www.ds.eng.monash.edu.au/techrep/reports/2004/MECSE-6-2004.pdf
[41] http://dapissarenko.com/resources/2002_12_01_eigenfaces.pdf
[42] Turk, M. and A. Pentland, Eigenfaces for Recognition. Journal of Cognitive Neuroscience, 1991. **3**(1): p. 71-86.
[43] http://www.cse.iitb.ac.in/~adityak/seminar.pdf
[44] L. I. Smith. A tutorial on principal components analysis, February 2002.URL http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf.
[45] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991a. URL http://www.cs.ucsb.edu/~mturk/Papers/jcn.pdf.
[46] http://www.cs.iastate.edu/~alex/classes/2007_Spring_575X/slides/13_EigenFaces.pdf
[47] http://www.cse.iitb.ac.in/~adityak/seminar.pdf

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

[48] http://dapissarenko.com/resources/2002_12_01_eigenfaces.pdf
[49] Multi-lane architecture for eigenface based real-time face recognition Rajkiran Gottumukkal, Hau T. Ngo, Vijayan K. Asari, Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA 23529, USA], [http://www.cse.iitb.ac.in/~adityak/seminar.pdf
[50] http://www.omg.org/gettingstarted/what_is_uml.htm
[51] http://www.ilstu.edu/~asharm4/432/432.ppt#270,7,Slide 7
[52] http://www.physics.emory.edu/~weeks/graphics/mkppm.html
[53] http://schmidt.devlib.org/java/pixel-image-io-libraries.html
[54] http://www.cs.ucsb.edu/~mturk/Papers/mturk-CVPR91.pdf
[55] http://math.nist.gov/javanumerics/jama/
[56] http://en.wikipedia.org/wiki/Euclidean_distance
[57] http://www.netbeans.org/features/
[58] www.sun.com
[59] http://en.wikipedia.org/wiki/Classpath_(Java
[60] http://h10025.www1.hp.com/ewfrf/wc/document?lc=en&dlc=en&cc=us&docname=c00865850#N274
[61] http://java.sun.com/javase/technologies/desktop/media/jmf/reference/api/index.html
[62] http://java.sun.com/j2se/1.3/docs/guide/2d/apijpeg/com/sun/image/codec/jpeg/package-summary.html#package_description
[63] http://math.nist.gov/javanumerics/jama/doc/
[64] [http://www.ilkeratalay.com/download/eigenfaces_msc_thesis.pdf].