HOME AUTOMATION USING FACE AS WELL AS FACIAL EXPRESSION RECOGNITION

Dissertation submitted in fulfillment of the requirements for the Degree of

BACHELORS OF TECHNOLOGY

By

121037-Vani Deeppak

121042-Aayshu Rani

121047-Dimple Thakur

Under the guidance of- Ms. Pragya Gupta

Department of Electronics and Communications

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

June 2016

TABLE OF CONTENTS

<u>DECLARATION BY THE SCHOLAR</u>

We hereby declare that the work reported in the B-Tech thesis entitled **"Home Automation Using Face as well as Facial expression Recognition"** submitted at **Jaypee University of Information Technology, Waknaghat India,** is an authentic record of our work carried out under the supervision of **Ms. Pragya Gupta**. We have not submitted this work elsewhere for any other degree or diploma.

_____________ ____________ ______________

Vani Deeppak Aayshu Rani Dimple Thakur

Department of Electronics and Communications,

Jaypee University of Information Technology, Waknaghat , India.

Date:

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the B-Tech. thesis entitled "**Home Automation Using Face as well as Facial Expression Recognition**", submitted by **Vani Deeppak, Aayshu Rani, Dimple Thakur** at **Jaypee University of Information Technology, Waknaghat**, **India,** is a bonafide record of their original work carried out under my supervision. This work has not been submitted elsewhere for any other degree or diploma.

____

Ms. Pragya Gupta

Jaypee University of Information Technology, Waknaghat, India

Date:

ACKNOWLEDGEMENT

The sense of fulfillment that comes along with the successful completion of any task is incomplete without expressing gratitude to all those who made it possible with their guidance and support.

We thank the almighty for giving us such excellent facilities and support through the way of **JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY** for the completion of this project.

We would like to express sincere gratitude to our **Head of Department**, **Prof. S.V Bhooshan** for having extended all department facilities without hesitation.

We especially thank **Ms Pragya Gupta**, our **project guide** for giving us extremely valuable guidance and support throughout the course of the project.

We also thank all the staff members and especially Mr. Mohan Sharma of **Electronics and Communication Engineering Department** for their kind co-operation and timely help.

ABSTRACT

The project has been divided into 3 phases:

- Face recognition
- Facial expression recognition
- Home automation
  - Music and light arrangement control
  - Automatic door control system

The goal is to implement the system (model) for a particular face and distinguish it from a large number of stored faces with some real-time variations as well. The Eigen face approach uses Principal Component Analysis (PCA) algorithm for the recognition of the images. Furthermore, after detection and recognition of the face we aim to control the opening and closing of door and also with the detection of facial expressions we aim to control the music and lighting arrangements of a room.

The aim of this project also is to help us for improvement of the door security of sensitive locations by using face detection and recognition.

The approach using Eigen faces and PCA is quite robust in the treatment of face images with varied facial expressions as well as the directions. However, this approach is sensitive to images with uncontrolled illumination conditions. One of the limitations for Eigen face approach is in the treatment of face images with varied facial expressions and with glasses.

LIST OF ACRONYMS & ABBREVIATIONS

ICA	Independent Component Analysis
LDA	Linear Discriminant Analysis
PCA	Principal Component Analysis
NMF	Non negative Matrix Formulation
2DPCA	2 Dimensional Principal Component Analysis
EDC	Euclidean Distance Classifier
GUI	Graphic User Interface
IC	Integrated Circuits
DC	Direct Current

LIST OF FIGURES

PART I-FACE RECOGNITION

CHAPTER 1

INTRODUCTION

Humans have always had the innate ability to acknowledge and distinguish between faces, yet computers recently have shown the same ability. In the mid Nineteen Sixties, scientists began work on using the laptop to acknowledge human faces. Since then, facial recognition software has developed and come back with extended means. Face recognition is one of the foremost successful applications of image analysis and understanding and has gained a lot of attention in recent years. Research shows that no specific projection–metric combination is the best across all commonplace tests and also the selection of acceptable projection–metric combination will solely be done for a selected task. Over the last ten years or so, face recognition has become one of the most popular areas of research in computer vision and one of the most successful applications of image analysis and understanding.

In general, face recognition techniques can be divided into two groups based on the face representation they use:

1. Appearance-based, which uses holistic texture features and is applied to either whole-face or specific regions in a face image;
2. Feature-based, this uses geometric facial features (mouth, eyes, brows, cheeks etc.) and geometric relationships between them [1].

1.1. Subspace Analysis

Appearance-based subspace analysis, one of the oldest, among many approaches to the problem of face recognition, gives the most promising results.

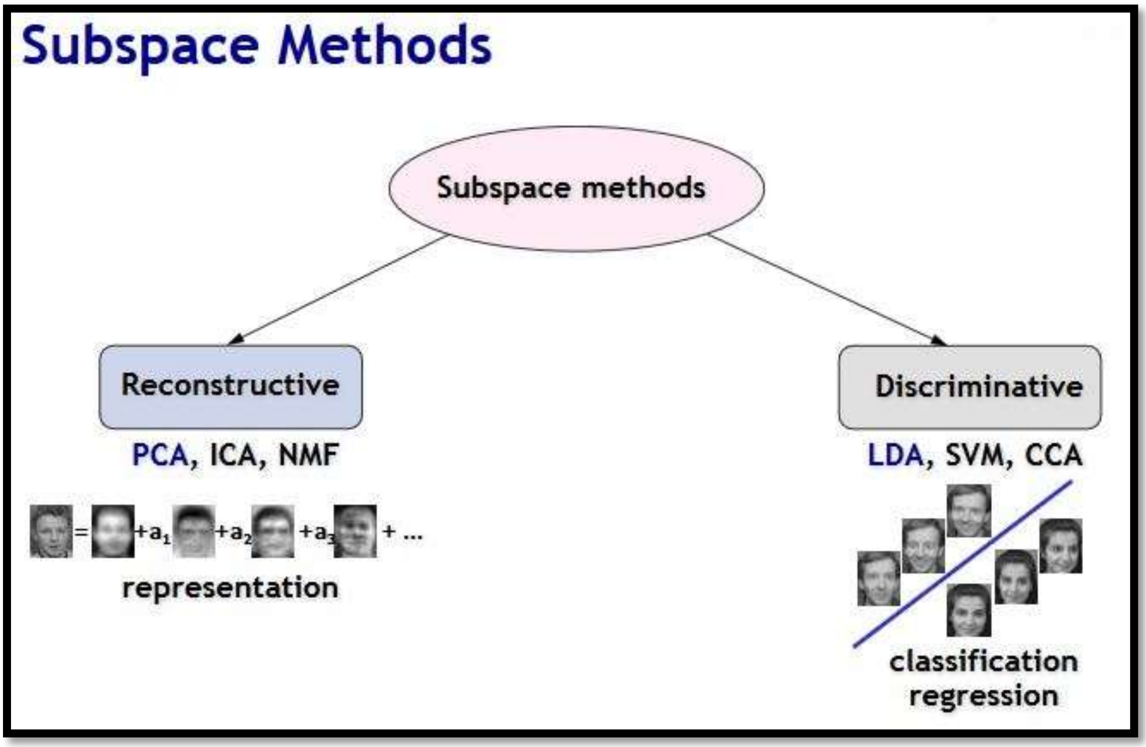

Figure 1.1 Subspace Methods

Subspace is a subset of a larger space and is such that it also contains the properties of the larger space. Generally the images are high dimensional in nature. A 2D image I with m rows and n columns can be viewed as a vector. Since space derived this way is highly dimensional since 'mxn' is very large, recognition in it is not possible. Thus, recognition algorithms usually obtain lower dimensional spaces to do the actual recognition while retaining as much information possible from the initial images as possible.

In short, Subspace analysis is a method, which is done by projecting an image into a lower dimensional space (subspace) and after that recognition is performed by measuring the distances between known images and the image to be recognized.

The two types of subspace methods are as follows:

- Linear or reconstructive subspace methods that provide sufficient reconstruction of the data, offer an efficient way of dealing with missing pixels, outliers, and occlusions that often appear in the visual data.
- Discriminative methods, on the other hand, are better suited for classification tasks, are highly sensitive to corrupted data.

The most challenging part of such a system is finding an adequate subspace.

Projection methods to be presented are:

Principal Component Analysis (PCA)	Independent Component Analysis (ICA)	Linear Discriminant Analysis (LDA)
PCA finds a set of the most representative projection vectors such that the projected samples retain most information about original samples.	ICA captures both second and higher order statistics and projects the input data onto the basis vectors that are as statistically independent as possible	LDA uses the class information and finds a set of vectors that maximize the between-class scatter while minimizing the within-class scatter.

All three projection methods are so called subspace analysis methods.

1.2. What are Eigen Faces?

Eigen faces is the name given to a set of eigenvectors when they are used in the computer vision problem of human face recognition.[6] The approach of using eigen faces for recognition was developed by Sirovich and Kirby (1987) and used by Matthew Turk and Alex Pentland in face classification [7]. The eigenvectors are derived from the covariance matrix of the probability distribution over the high-dimensional vector space of face images. The eigen faces themselves form a basis set of all images used to construct the covariance matrix. This produces dimension reduction by allowing the smaller set of basis images to represent the original training images. Classification can be achieved by comparing how faces are represented by the basis set.

A set of eigen faces can be produced by performing a mathematical process called principal component analysis (PCA) on a large set of images depicting different human faces. Informally, eigen faces can be considered a set of "standardized face ingredients", derived from statistical analysis of several pictures of faces. Any human face can be considered to be a combination of these standard faces. For example, one's face might be composed of the average face plus 10% from eigen face 1.55% from eigen face 2, and even -3% from eigen face 3. Remarkably, it does not take many eigen faces combined together to achieve a fair approximation of most faces. Also, because a person's face is recorded by just a list of values (one value for each eigen face in the database used) and not as a digital photograph, hence much less space is taken for each person's face. The eigen faces that are created will appear as light and dark areas that are set in a specific way.

1.3. Computing the eigenvectors

Performing PCA directly on the covariance matrix of the images is often computationally infeasible. If small, say 100×100, greyscale images are used, each image is a point in a 10,000-dimensional space and the covariance matrix **S** is a matrix of $10{,}000 \times 10{,}000 = 10^8$ elements. However the rank of the covariance matrix is limited by the number of training examples: if there are N training examples, there will be at most $N - 1$ eigenvectors with non-zero eigenvalues. If the number of training examples is smaller than the dimensionality of the images, the principal components can be computed more easily as follows.

Let **T** be the matrix of preprocessed training examples, where each column contains one mean-subtracted image. The covariance matrix can then be computed as $\mathbf{S} = \mathbf{T}\mathbf{T}^T$ and the eigenvector decomposition of **S** is given by:

$$\mathbf{S}\mathbf{v}_i = \mathbf{T}\mathbf{T}^T\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

However $\mathbf{T}\mathbf{T}^T$ is a large matrix, and if instead we take the eigen value decomposition of

$$\mathbf{T}^T\mathbf{T}\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

then we notice that by pre-multiplying both sides of the equation with **T**, we obtain

$$\mathbf{T}\mathbf{T}^T\mathbf{T}\mathbf{u}_i = \lambda_i \mathbf{T}\mathbf{u}_i$$

Meaning that, if $\mathbf{u}_i$ is an eigenvector of $\mathbf{T}^T\mathbf{T}$, then $\mathbf{v}_i = \mathbf{T}\mathbf{u}_i$ is an eigenvector of **S**. If we have a training set of 300 images of 100×100 pixels, the matrix $\mathbf{T}^T\mathbf{T}$ is a 300×300 matrix, which is much more manageable than the $10{,}000 \times 10{,}000$ covariance matrix. Notice however that the resulting vectors $\mathbf{v}_i$ are not normalized; if normalization is required it should be applied as an extra step [8].

1.4. Use in facial recognition

Facial recognition was the source of motivation behind the creation of eigenfaces. For this use, eigen faces have advantages over other techniques available, such as the system's speed and efficiency. As eigen face is primarily a dimension reduction method, a system can represent many subjects with a relatively small set of data. As a face recognition system it is also fairly invariant to large reductions in image sizing, however it begins to fail considerably when the variation between the seen images and probe image is large.

To recognize faces, gallery images, those seen by the system, are saved as collections of weights describing the contribution each eigen face has to that image. When a new face is presented to the system for classification, its own weights are found by projecting the image onto the collection of eigen faces. This provides a set of weights describing the probe face. These weights are then classified against all weights in the gallery set to find the closest match. A nearest neighbour method is a simple approach for finding the Euclidean Distance between two vectors, where the minimum can be classified as the closest subject.

Intuitively, recognition process with eigen face method is to project query images into the face-space spanned by eigen faces we have calculated and in that face-space find the closest match to a face class.

Pseudo code:

- Given input image vector $U \in \Re^n$, the mean image vector from the database M, calculate the weight of the kth eigenface as:

$$w_k = V_k^T(U - M)$$

Then form a weight vector $W = [w_1, w_2, ..., w_k, ..., w_n]$

- Compare W with weight vectors W_m of images in the database. Find the Euclidean distance.

$$d = ||W - W_m||^2$$

- If $d < \epsilon_1$, then the m^th entry in the database is a candidate of recognition.
- If $\epsilon_1 < d < \epsilon_2$, then *U* may be an unknown face and can be added to the database.
- If $d > \epsilon_2, U$ is not a face image.

The weights of each gallery image only convey information describing that image, not that subject. An image of one subject under frontal lighting may have very different weights to those of the same subject under strong left lighting. This limits the application of such a system. Experiments in the original Eigen face paper presented the following results: an average of 96% with light variation, 85% with orientation variation, and 64% with size variation. Various extensions have been made to the eigen face method such eigen features. This method combines facial metrics (measuring distance between facial features) with the eigen face representation. Another method similar to the eigen face technique is 'fisher faces' which uses linear discriminant analysis. This method for facial recognition is less sensitive to variation in lighting and pose of the face than using eigen faces. Fisher face utilizes labeled data to retain more of the class specific information during the dimension reduction stage [8].

CHAPTER 2

METHODS FOR ANALYSIS

A number of current face recognition algorithms use face representations found by unsupervised statistical methods. Typically these methods find a set of basis images and represent faces as a linear combination of those images.

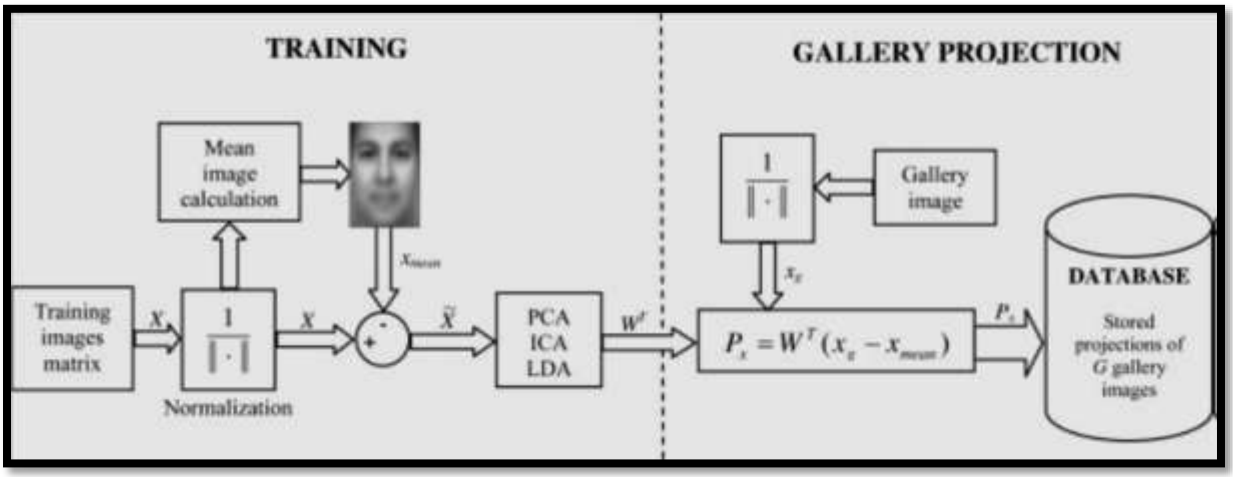

Figure 2.1 Illustration of subspace appearance-based face recognition system

Three principal methods are discussed as follows:

2.1. Principal Component Analysis (PCA): Given an s-dimensional vector representation of each face in a training set of M images, PCA tends to find a t-dimensional subspace whose basis vectors correspond to the maximum variance direction in the original image space. This new subspace is normally lower dimensional. New basis vectors define a subspace of face images called face space. All images of known faces are projected onto the face space to find sets of weights that describe the contribution of each vector.

To identify an unknown image, that image is projected onto the face space as well to obtain its set of weights. By comparing a set of weights for the unknown face to sets of weights of known faces, the face can be identified. If the image elements are considered as random variables, the PCA basis vectors are defined as eigenvectors of the scatter matrix ST defined as:

$$S_T = \sum_{i=1}^{M} (x_i - \mu) \cdot (x_i - \mu)^T$$

Where μ is the mean of all images in the training set and x_i is the i^{th} image with its columns concatenated in a vector. The projection matrix WPCA is composed of t eigenvectors corresponding to t largest eigen values, thus creating a t-dimensional face space. Since these eigenvectors (PCA basis vectors) look like some ghostly faces they were conveniently named eigen faces.

2.2. Independent Component Analysis (ICA): PCA considered image elements as random variables with Gaussian distribution and minimized second-order statistics. Clearly, for any non-Gaussian distribution, largest variances would not correspond to PCA basis vectors. Independent Component Analysis (ICA) minimizes both second-order and higher order dependencies in the input data and attempts to find the basis along which the data (when projected onto them) are statistically independent. provided two architectures of ICA for face recognition task: Architecture I – statistically independent basis images (ICA1 in our experiments) and Architecture II – factorial code representation (ICA2 in our experiments). PCA is used to reduce dimensionality prior to performing ICA.

2.3. Linear Discriminant Analysis (LDA): Linear Discriminant Analysis (LDA) finds the vectors in the underlying space that best discriminate among classes. For all samples of all classes the between-class scatter matrix S_B and the within-class scatter matrix S_W are defined by:

$$S_B = \sum_{i=1}^{c} M_i \cdot (x_i - \mu) \cdot (x_i - \mu)^T$$

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i) \cdot (x_k - \mu_i)^T$$

Where M_i is the number of training samples in class i, c is the number of distinct classes, μ_i is the mean vector of samples belonging to class i and x_i represents the set of samples belonging to class i with x_k being the k^{th} image of that class. S_W represents the scatter of features around the mean of each face class and S_B represents the scatter of features around the overall mean for all face classes.

The goal is to maximize S_B while minimizing S_W, in other words, maximize the ratio det|SB|/det|Sw|.

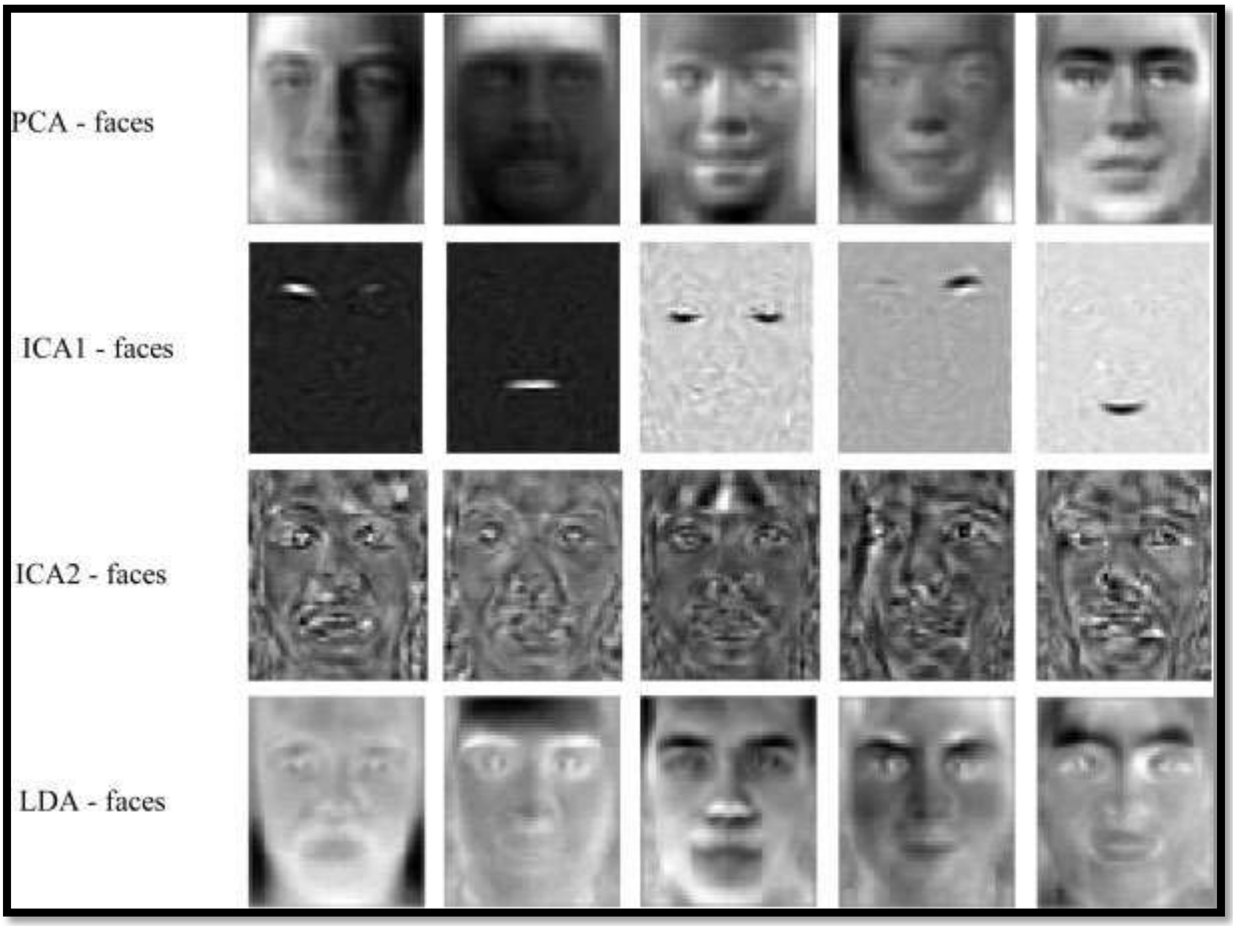

Figure 2.2 Face representations found by PCA, ICA1, ICA2, LDA

In Figure 2.2 PCA-faces (eigen faces), ICA1-faces, ICA2-faces, and LDA-faces can be seen. These ghostly faces are basis vectors produced by projection methods, reshaped to a matrix form of the original image size for convenience. This is a good illustration of the differences between subspaces derived by each of those projection methods. If we take a closer look at the basis vector representations it can be seen that PCA, LDA, and

ICA2 produce global features; every image feature is influenced by every pixel. ICA1 produces spatially localized features that are only influenced by small parts of an image, thus isolating particular parts of faces. Logical conclusion is that ICA1 should be optimal for recognizing facial actions and suboptimal for recognizing temporal changes in faces or images taken under different illumination conditions [2].

CHAPTER 3

METHODOLOGY

Step-1: Firstly, the image matrix I of size (N x N) pixels is converted to the image vector Γ of size (P x1) where P = (N x N). Training Set: $\Gamma = [\Gamma_1\ \Gamma_2\ \ldots\ \Gamma_N]$

Step-2: Average face image is calculated by $\Psi = (1/M) \sum \Gamma_i$ i=1 to M

Each face differs from the average by $\Phi_i = \Gamma_i - \Psi$

Difference Matrix: $A = [\Phi_1\ \Phi_2\ \ldots\ \Phi_M]$

Step-3: A covariance matrix is constructed as: $C = AA^T$, where size of C is (P×P).

1) This covariance matrix is very hard to work with due to its huge dimension that causes computational complexity.

2) The covariance matrix with reduced dimensionality is $L = A^TA$, where size of L is (M×M).

Step-4: A face image can be projected into this face space by $\Omega_k = U_k{}^T\Phi_i$

Step-5: Test image vector: Γ_t Mean subtracted image vector: $\Phi_t = \Gamma_t - \Psi$ the test image is projected into the face space to obtain a vector: $\Omega = U_k{}^T\Phi_t$

Step-6: Finding the minimum distance between the test image and the training images. The face with minimum Euclidian distance shows the similarity to test image. The distance of test image Ω to each training image is called Euclidean distance and is defined by, $\varepsilon_k^2 = \| \Omega - \Omega_k \|^2$

By choosing a threshold value Θ that is the maximum acceptable value for known images and comparing it with the minimum, test image can be recognized as known or unknown face image.

$$\begin{cases} \text{If } \varepsilon_k(\min) \geq \Theta \text{, the test image is recognized an unknown face.} \\ \text{If } \varepsilon_k(\min) < \Theta \text{, the test image is a known face [3].} \end{cases}$$

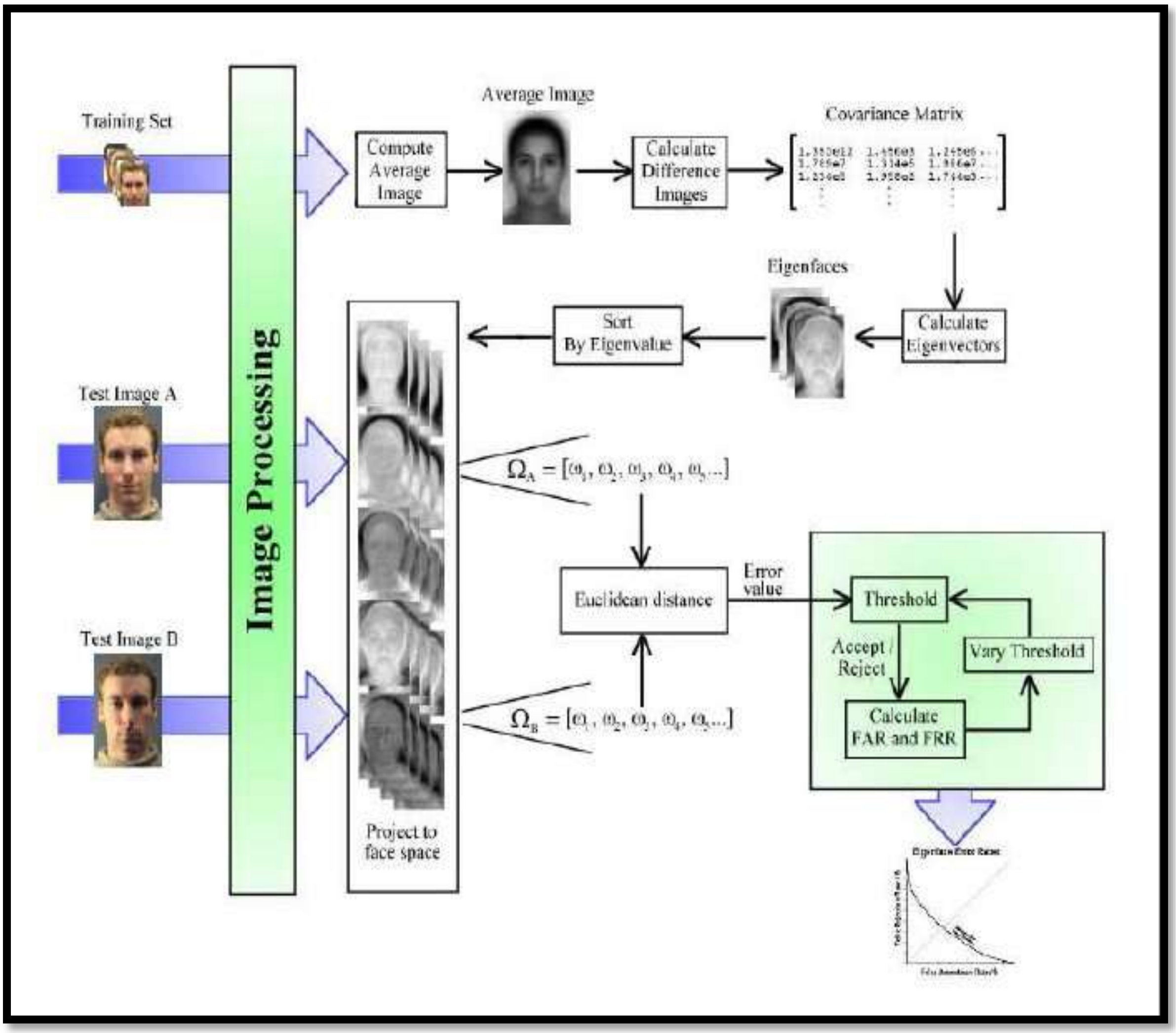

Figure 3.1 Procedure for recognition system

3.1 Implementation

We created GUI for facial recognition system (all the sub parts are the functions) the photos for which are also shown:

1) Create data base of faces –

    Add image

    Add folder

    Exit

2) Delete database

    Delete image

Delete folder

Exit

3) Train system

4) Face recognition

Input image from file

Capture now

Recognition

Exit

5) Exit

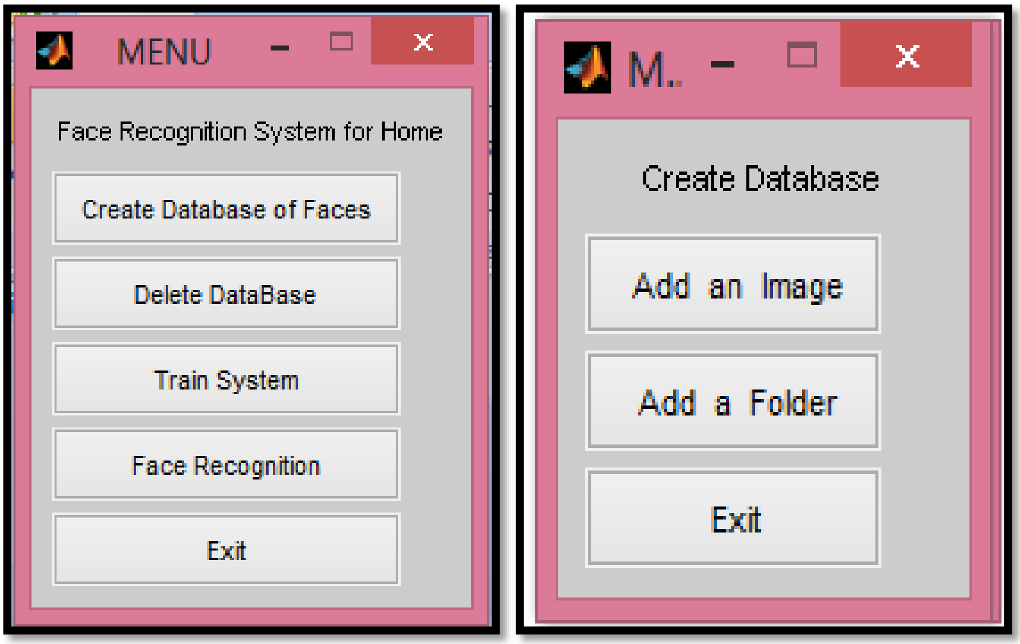

Figure 3.2 Main Menu and the First sub-part of main menu

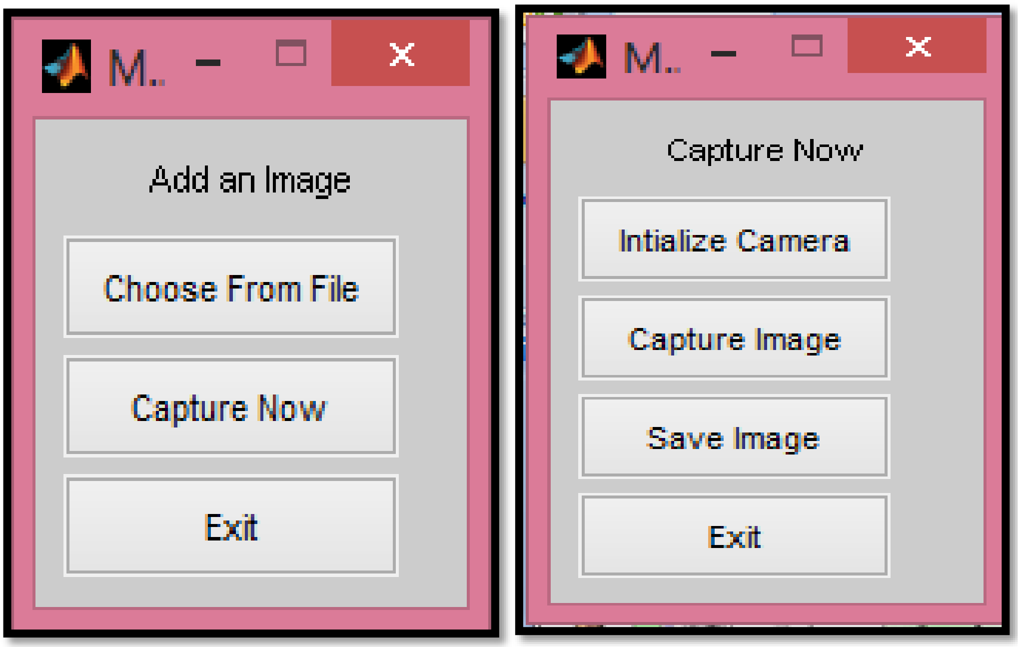

Figure 3.3 GUI for Add an Image and Capture Image

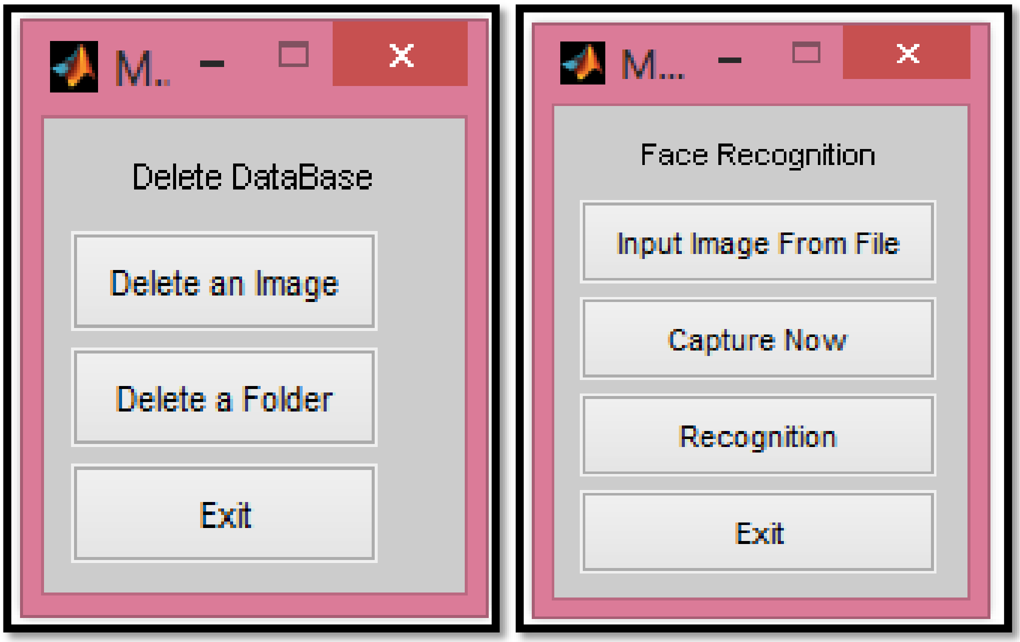

Figure 3.4 GUI for deleting database and face recognition

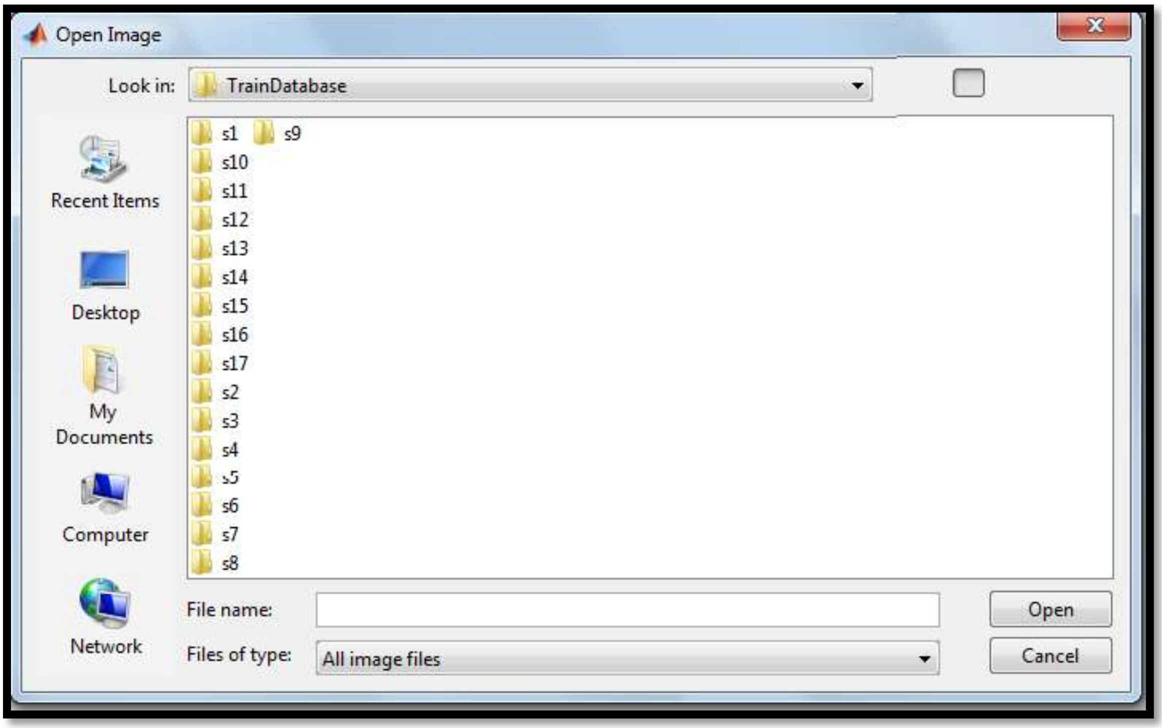

Figure 3.5 Training Data set (a)

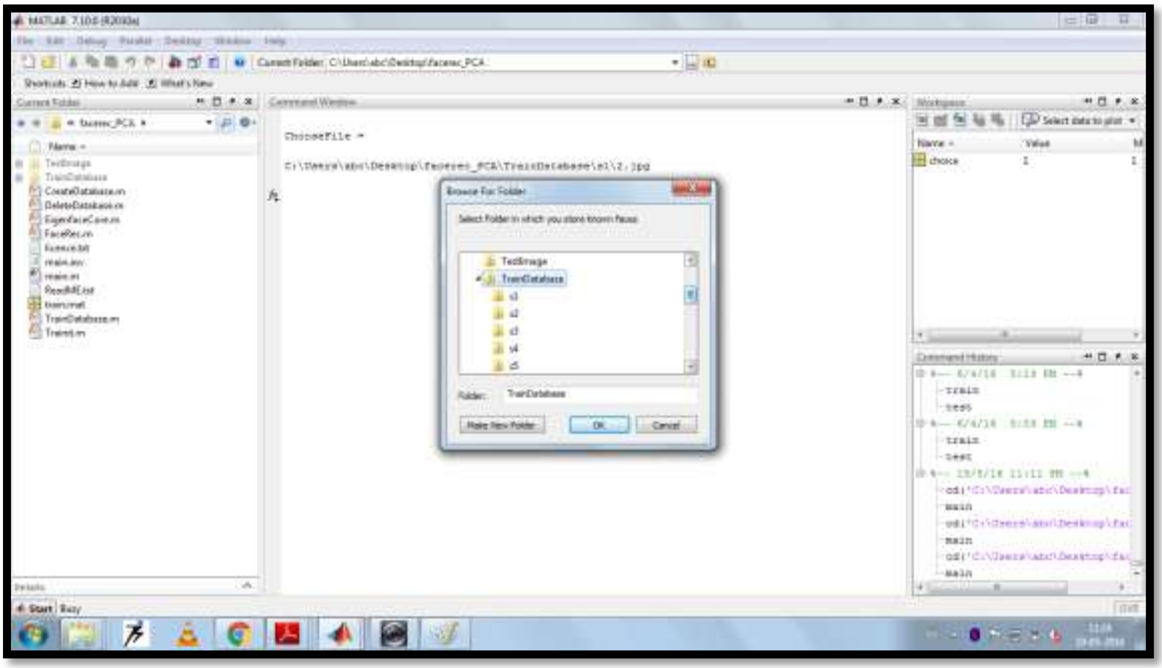

Figure 3.6 Training Data set (b)

Here the database of individuals is created by adding the images as shown in figure 3.4.The Train Database folder contains all the subfolders named as s1,s2,------,sn where 'si' is the folder for one person as shown in figure 3.8.Now the test image is trained.

Then face recognition is done. If the test image matches any of the images in the subfolders of Train database then a window appears which shows both the test image as well as the equivalent image as well as name of the respective folder as shown in figure 4.1 and figure 4.2.

CHAPTER 4

RESULTS

As shown in Figure 4.1, the test image matches with the image which was present in the database prior to the facial detection and recognition.

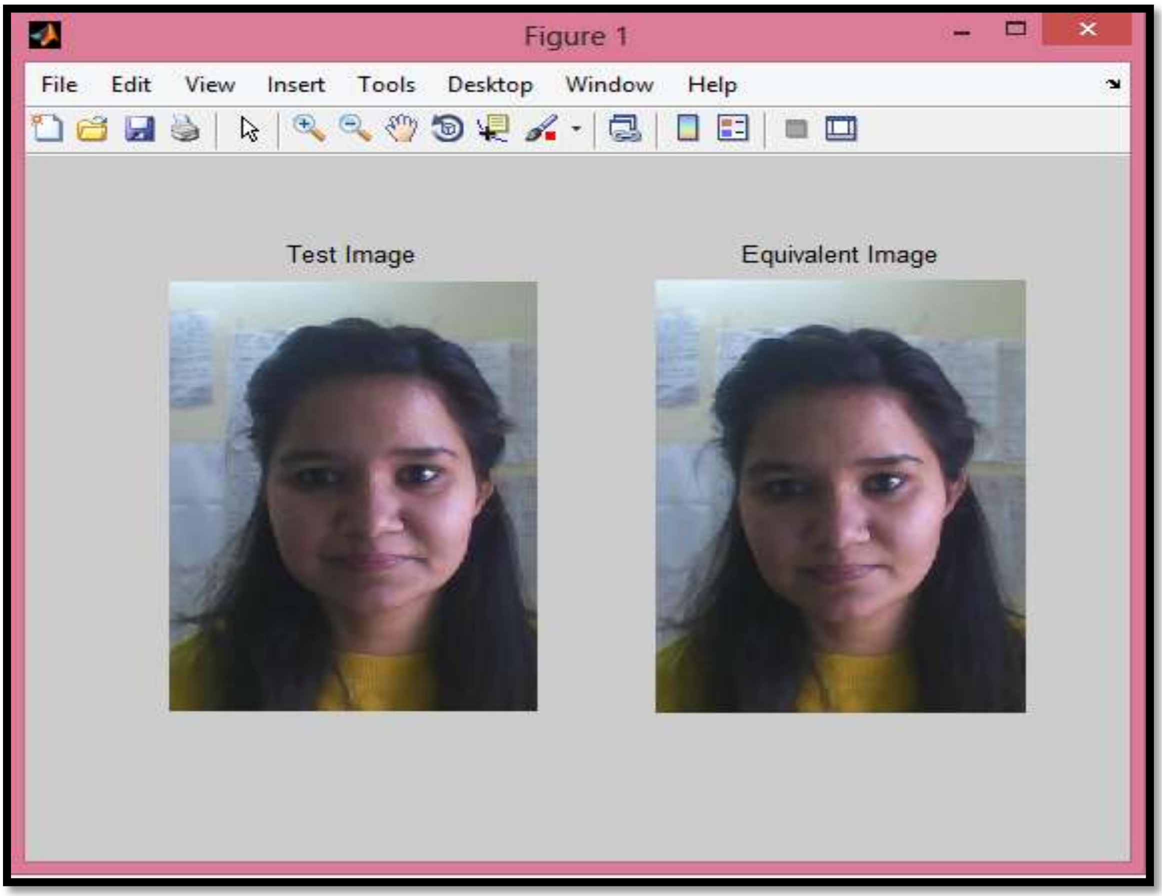

Figure 4.1 Result obtained

We also see that our system not only detects and recognizes the face but also indicates the person number from the trained data set as already explained in section 3.1.

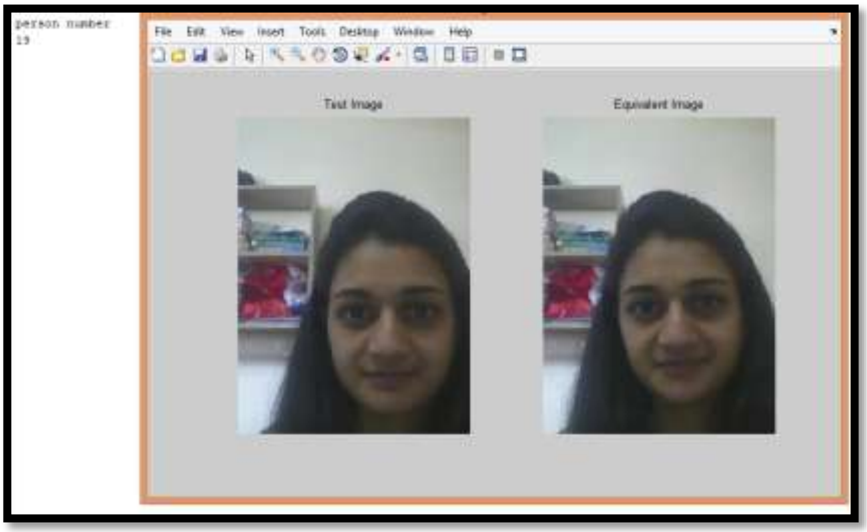

Figure 4.2 Result indicating the person number from training set

PART II-FACIAL EXPRESSION RECOGNITION

CHAPTER 5

INTRODUCTION

Human faces are complex set of objects with several features that can dissent all the time based on influence of external environment. Since expressions are the most vital mode of non-verbal communication between people. Hence with the help of expressions, detection of human emotions becomes easier by analyzing human faces Facial expression carries crucial information about the mental, emotional and even physical states of the conversation. Facial expression recognition has broad application prospects, such as user-friendly interface between man and machine like emotional robot etc. With facial expression recognition systems, the machine like computers will be able to assess the human expressions depending on their state in the same way that human do.

Due to technological advancements; there is an expectation of a world where humans and robots live together. The intelligent computers will be able to understand, interpret and respond to human intentions, emotions and moods. The facial expression recognition system applied in different areas of life such as security and surveillance, or in communications so as to make the answer machine more interactive with people. Moreover, it is powerful in signed language recognition system that deals with people that have problems with hearing and speaking. Recognizing the emotional state of a person by analyzing him/her facial expression appears challenging but is also essential.

However, humans have the natural and inbuilt ability to notice, recognize, differentiate and identify or pinpoint a person’s face. Expressions and feelings differs completely with different context. Facial Expression Recognition System recognizes human facial pictures containing some expression as input and categoryify it into acceptable expression class like happy, angry, fear, disgust, neutral, surprise, and sad. Expression detection is useful as a non-invasive methodology of lie detection and behaviour prediction. However, these facial expressions may be troublesome to notice to the undisciplined eye. In this paper we implements face expression recognition techniques with the help of Principal component analysis (PCA).

Below, in Figure 5.1 are seven common facial expressions shown of an individual [5].

Figure 5.1 Seven common facial expressions

CHAPTER 6

METHODOLOGY

6.1. Previous Approaches to Facial Expression Recognition

Bartlett explores and compares techniques for automatically recognizing facial actions in sequences of images. These techniques include analysis of facial motion through estimation of optical flow; holistic spatial analysis, such as independent component analysis, local feature analysis, and linear discriminant analysis; and methods based on the outputs of local filters, such as Gabor wavelet representations and local principal components. The system that recognizes various action units based on dense flow, feature point tracking and edge extraction. The system includes three modules to extract feature information: dense-flow extraction using a wavelet motion model, facial feature tracking, and edge and line extraction. The system that used colour information, proposes the use of non-negative matrix normalization (NMF) with colour cannel encoding [4]. This process is performed by representing the (RGB) colour channel as a three indexed data vector separately: red, green and blue channel for each image. Then the colour using non-negative matrix (NMF), a decoding method, is applied. This technique makes better use of the colour image because of the excessive iterative matrix and the decoding operation that involves inverting the matrix; the inherent processing cost was so big.

As opposed to Principal component analysis, two-dimensional principal component analysis (2DPCA) is based on 2D image matrices rather than 1D vector. In two dimensional Principal Component Analysis, Principal Component Analysis must be applied.

6.2. Approach Taken for Facial Expression Recognition

The work presented here provides a novel solution to the facial expression recognition problem, describing a facial recognition system that can be used in application of

Human computer interface. There are three main components to this system: a Feature Extraction, Principal Component Analysis and Euclidean Distance Classifier. To classify the images final facial expression recognition system uses Euclidean Distance Classifier (EDC). It recognizes expression of the seven basic emotions, namely happy, disgust, neutral, anger, sad, surprise and fear.

A. System Architecture

This section describes facial expression recognition system architecture. Our system is composed by four modules:

Pre-processing, Principal Component analysis and expression classification using Euclidian classifier represents the basic blocks of facial expression recognition system.

B. Pre-processing

Pre-processing is the next stage after entering the data into the facial expression recognition system. The important data that is needed for most facial expression recognition methods is face position. In pre-processing module images are resized from 256 x 256 pixel value to 280 x 180 pixel values.

C. Principal Component Analysis (PCA)

Principal component analysis is a standard technique used in the statistical pattern recognition and signal processing for data reduction.

- Gather x_i where i= 1 to p.
- Compute the mean m and subtract it to obtain xi-m.
- Compute the covariance matrix $C_{ij} = (x_i\text{-}m)\ (x_i\text{-}m)\ T$.
- Determine Eigen values and Eigenvectors of covariance matrix C such that CV=AV where A=dig (λ_1, λ_2 ...λ_p.), a diagonal matrix is defined by the eigen values of the matrix C and V = (V_1,V_2 ....V_p) be the associated eigen vectors .such that $\lambda_1 \geq \lambda_2 \geq$,..........$\geq \lambda_p$.
- Select the first $l \leq p$ eigenvectors and discard p-l eigenvectors to find the data in new directions.
- If the orthogonal matrix contains the eigenvectors of C, then C can be decomposed as $C = VAV^T$ where A is diagonal matrix of eigen values. [9]

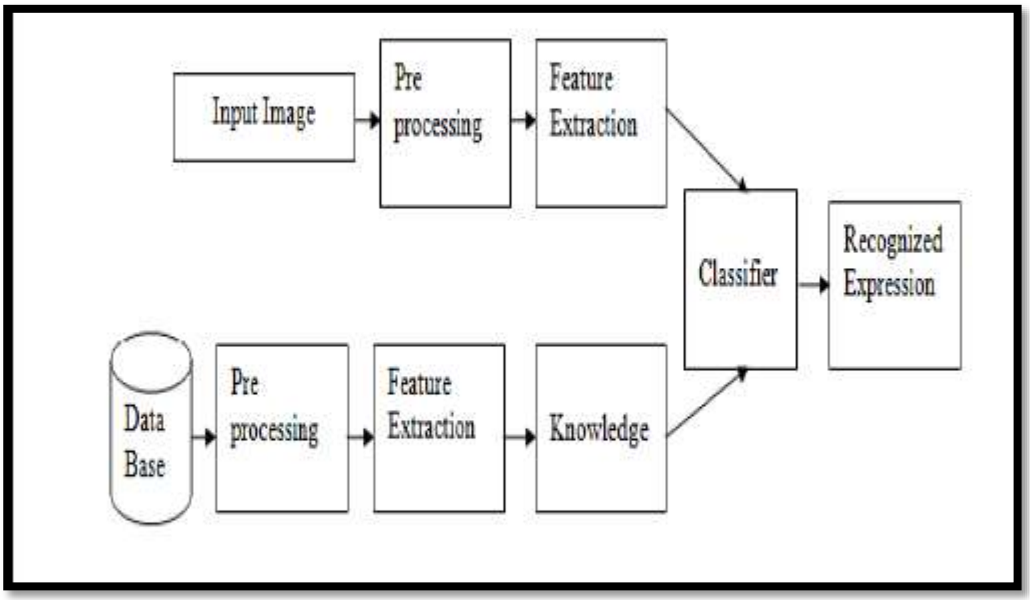

Figure 6.1 Facial Expression Recognition System Architecture

D. Facial Expression Classification

The proposed approach to the facial expression recognition involves following steps.

1) The train images are utilized to create a low dimensional face space. This is done by performing Principal Component Analysis in the training image set and taking the principal components with greater Eigen. In this process, projected versions of all the train images are also created.

2) The test images also projected on face space, all the test images are represented in terms of the selected principal components.

3) In order to determine the intensity of the particular expression its Euclidean distance from the mean of the projected neutral images is calculated.

4) The Euclidian distance of a projected test image from all the projected train images are calculated and the minimum value is chosen in order to find out the train image which is most similar to the test image.

5) The test image is assumed to fall in the same class that the closest train image belongs to.

CHAPTER 7

CONTROLLING MUSIC ARRANGEMENT OF THE ROOM USING FACIAL EXPRESSION RECOGNITION

7.1. Implementation

We worked on the music arrangement control of the room in which we build a customized system in which a person's choice of music is fed into the system, his various facial expressions recorded into the database known as the 'train images' as shown in figure 7.1. Once the data base is created facial expression recognition is done which after a recognizing a certain expression from the 'test images' and matching it to the database plays the desired song.

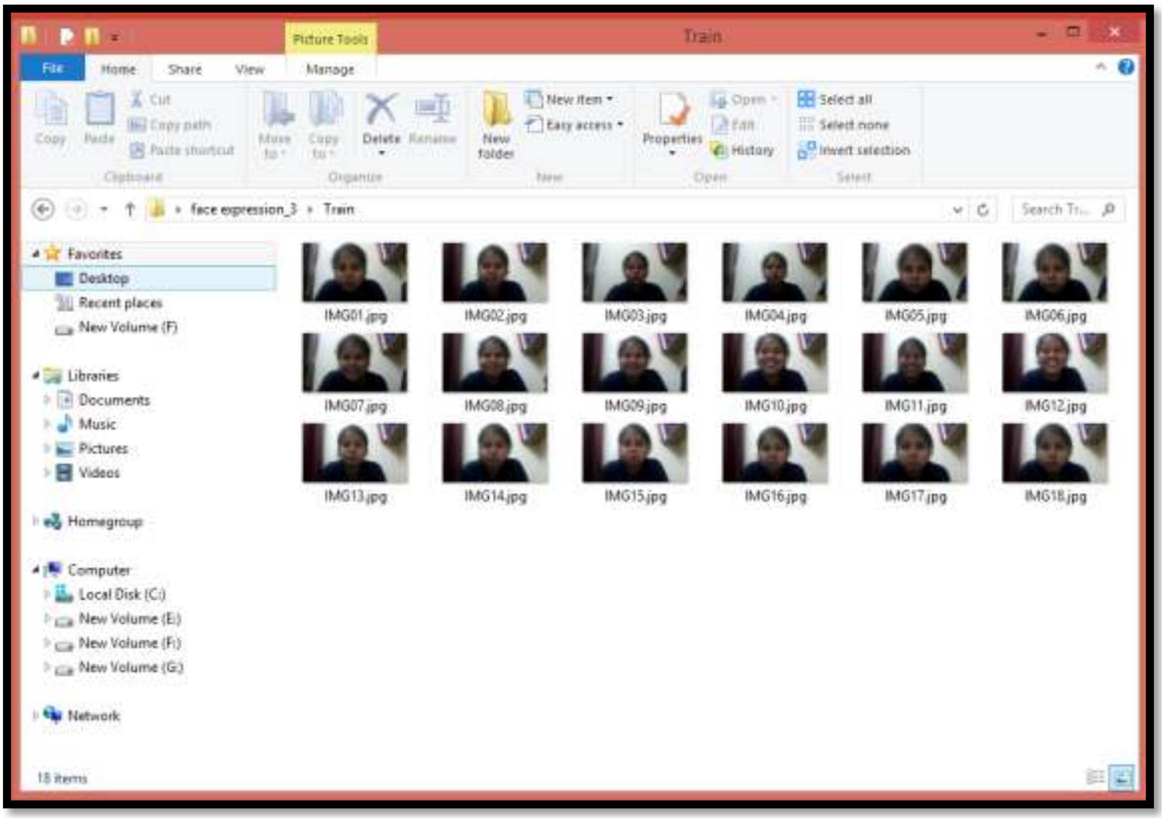

Figure 7.1 Screen-shot of Train Folder

7.2. Results

For our case, we used three basic expressions: Happy, Sad and Neutral. Every expression had six photos each in the train folder as shown in Figure 7.1. Three images of known expressions were put in the test folder as shown in Figure 7.2.

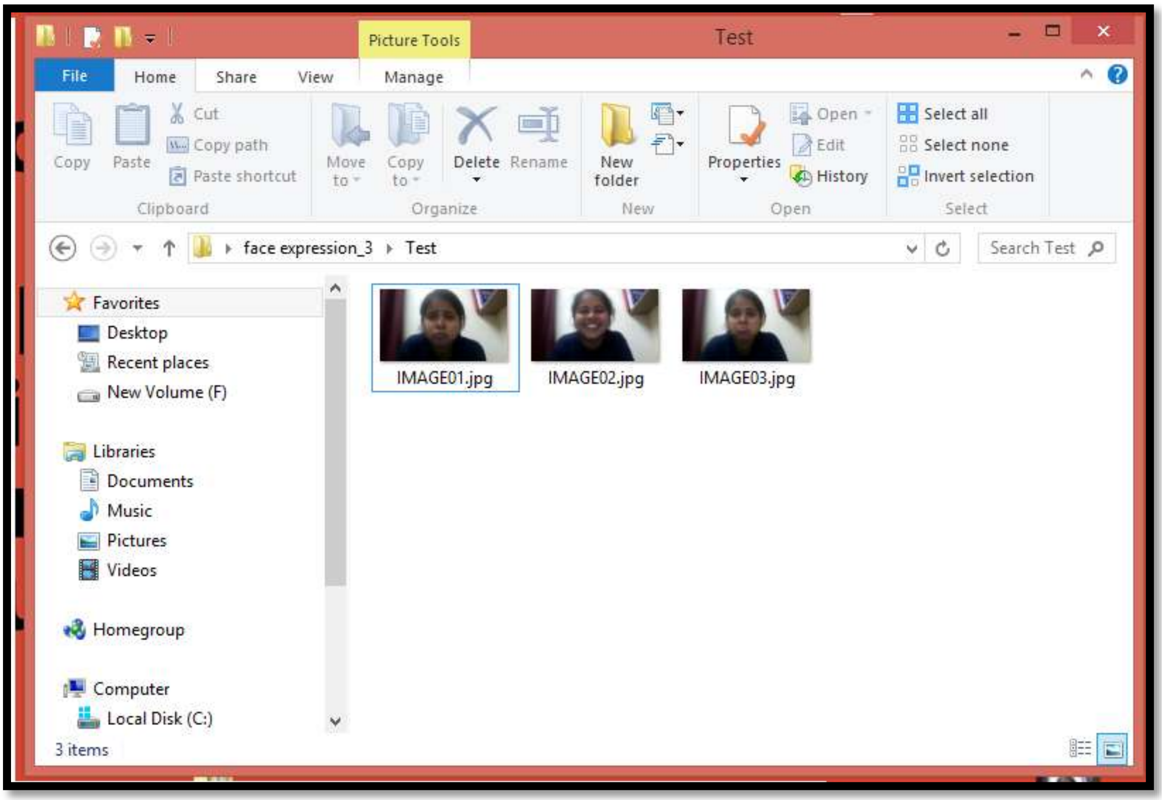

Figure 7.2 Screen-shot of Test Folder

According to each expression that was found out a song played. Since when a person is sad he/she would want to cheer up hence a cheerful song played and the contrary happened when that person was happy.

Slowly after all the images are processed according to the code, a result file is made of .txt extension as shown in Figure 7.3. Here as shown in the very same figure, each image put in the test folder was matched with a particular image in the train folder before the song for the particular expression played.

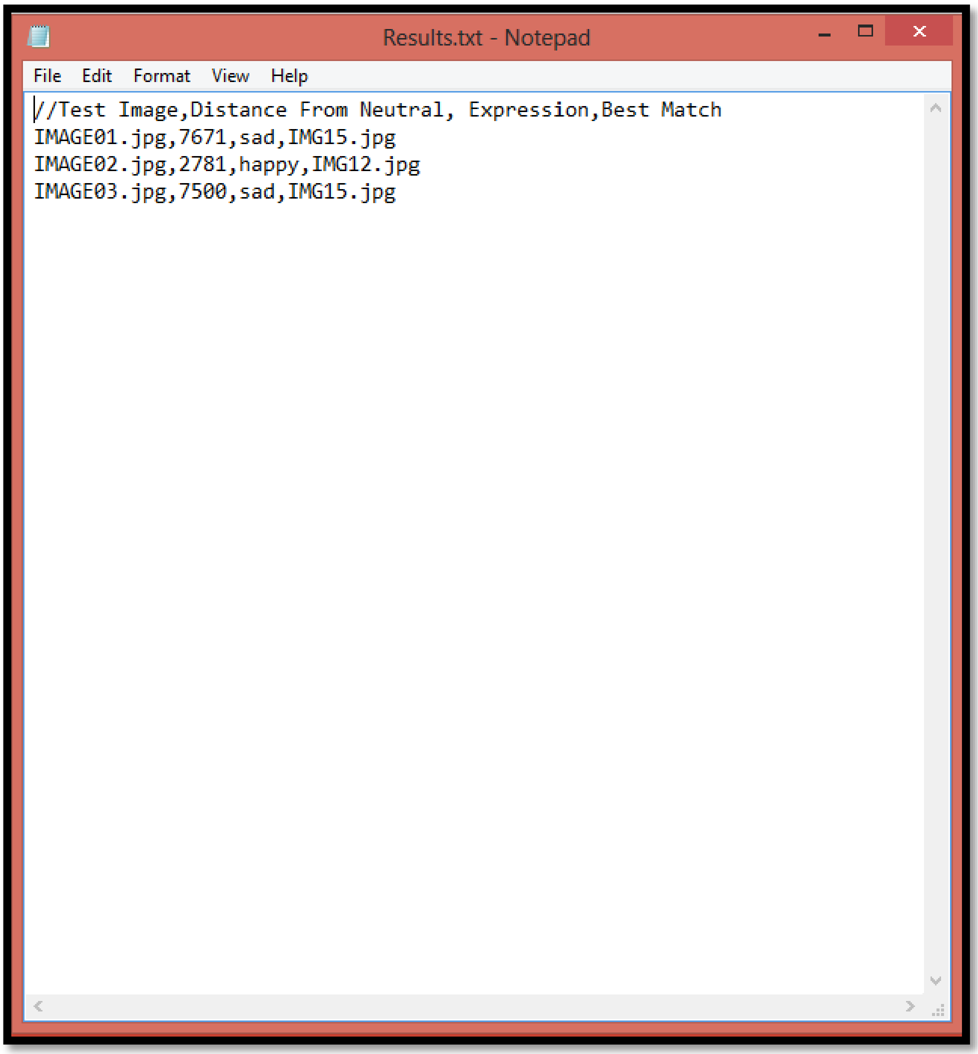

Figure 7.3 Result file on the notepad

PART III- AUTOMATIC DOOR CONTROL USING FACE DETECTION AND RECOGNITION

CHAPTER 8

INTRODUCTION

Most doors are controlled by persons with the use of keys, security cards, password or pattern to open the door. Face is a complex three-d structure and desires smart computing techniques for detection and recognition. Face Recognition on the basis of PCA is done .If a face is recognized, it is known, else it is unknown. The door will open mechanically for the noted person due to the command of the microcontroller. On the other hand, alarm will ring for the unknown person. Since PCA reduces the dimensions of face images while not losing vital options, facial images for several persons are often stored within the database.

Face detection is the start of the face recognition system. The performance of the entire face recognition system is influenced by the reliability of the face detection. By using face detection, it can establish solely the facial half of a picture despite the background of this image.

Face recognition commonly includes feature extraction, feature reduction and recognition or classification.PCA is an effective feature extraction methodology compounded on face as a worldwide feature. It reduces the dimension of images effectively and holds the primary info at constant time. In this paper, face recognition system is implemented using PCA formula/algorithm. Recognition or classification is done by the measure distance method such as Euclidean distance, which is used to classify the method such as Euclidean distance.

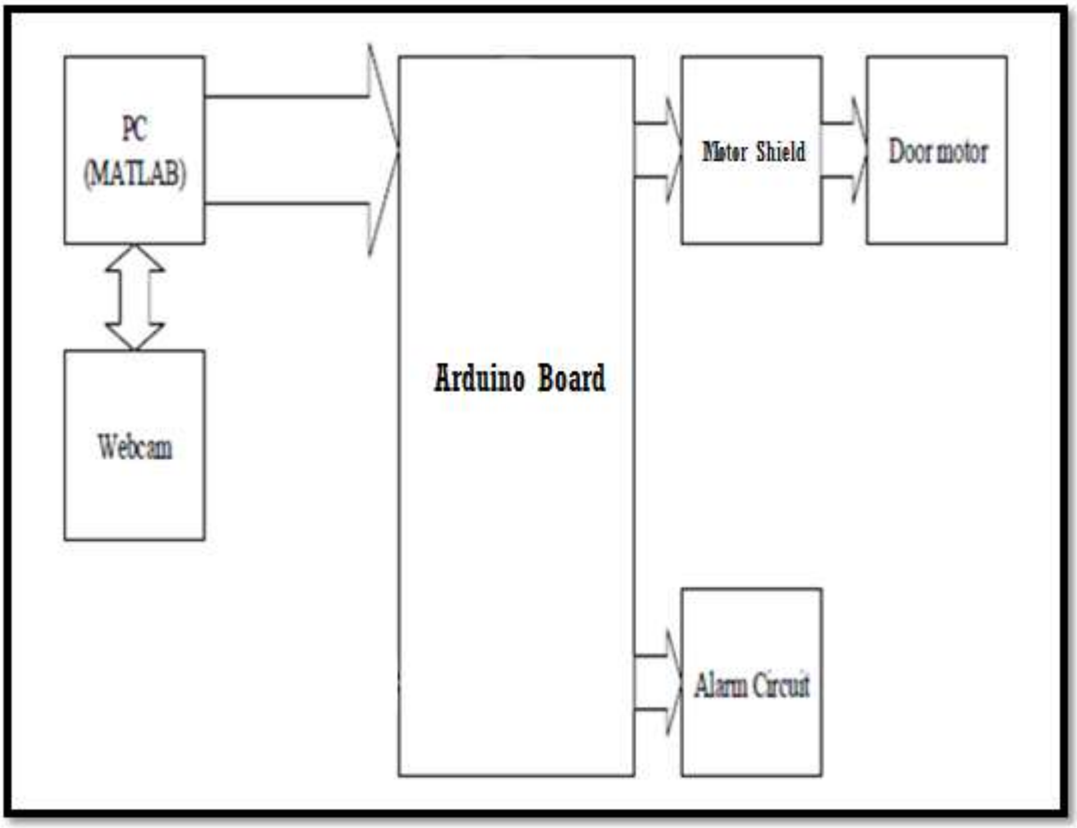

Figure 8.1 Block diagram of Automatic Door Control using face recognition

8.1. Working of automatic door control

It was created according the flowchart for the automatic door opening and closing system shown in Figure 8.2. Firstly, we assigned the required I/O pins and checked whether the data is received or not. When the received data is 'known', the motor is rotated in the forward direction for certain time. Motor is rotated in reverse direction after say 3 seconds. The motor is stopped after a certain time.But if the data is not known the buzzer will ring and the door will not open.

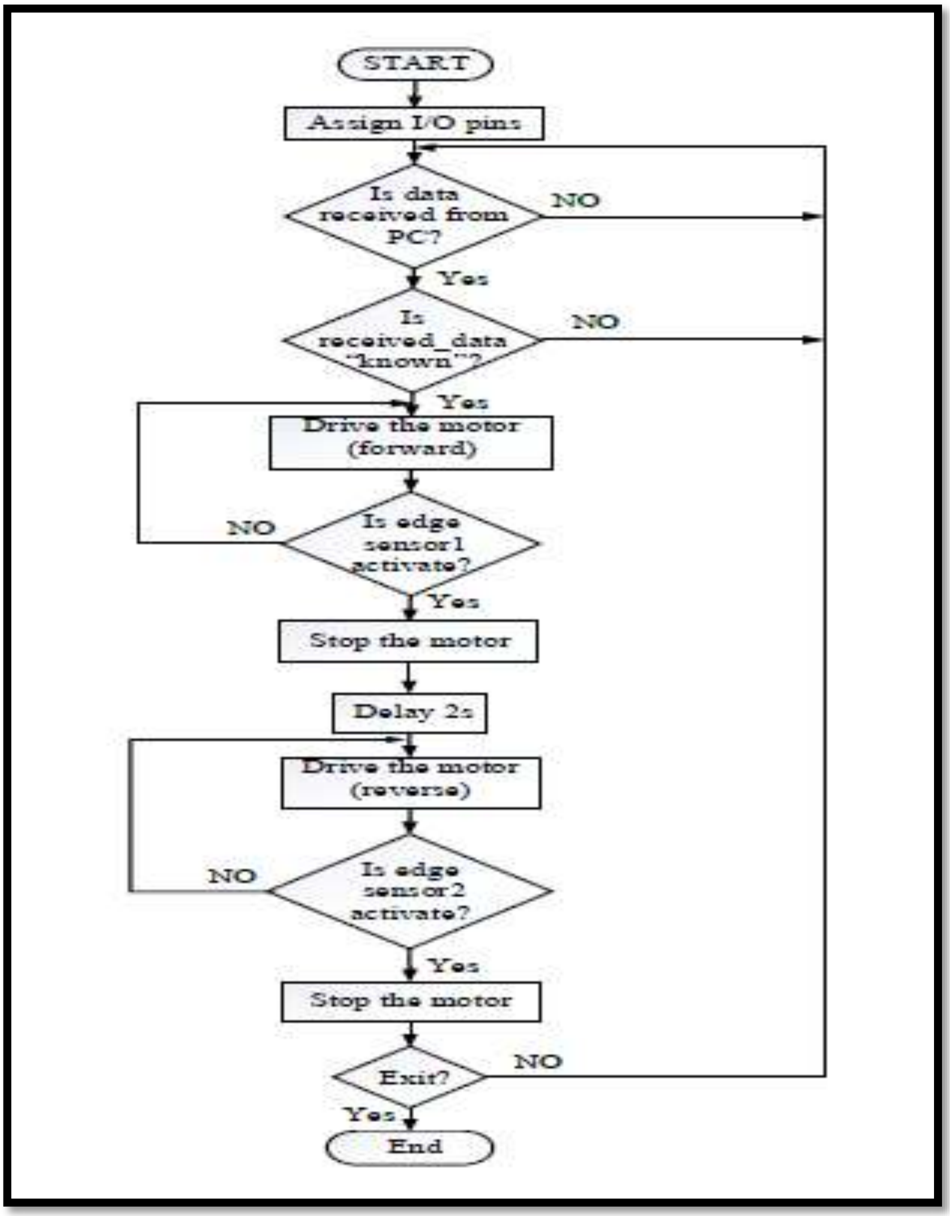

Figure 8.2 Flow Diagram of door control

8.2. Hardware used

1) ***Arduino***

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists and anyone interested in creating interactive objects or environments.

Below is pin diagram of Arduino Uno:

It has ATmega 328 microcontroller, a transmitter (TX), reciever (RX) for serial communication, External power supply, digital pins (2-13), analog pins (A0-A5), and a USB plug. Specifications for microcontroller are shown in Figure 8.4 .

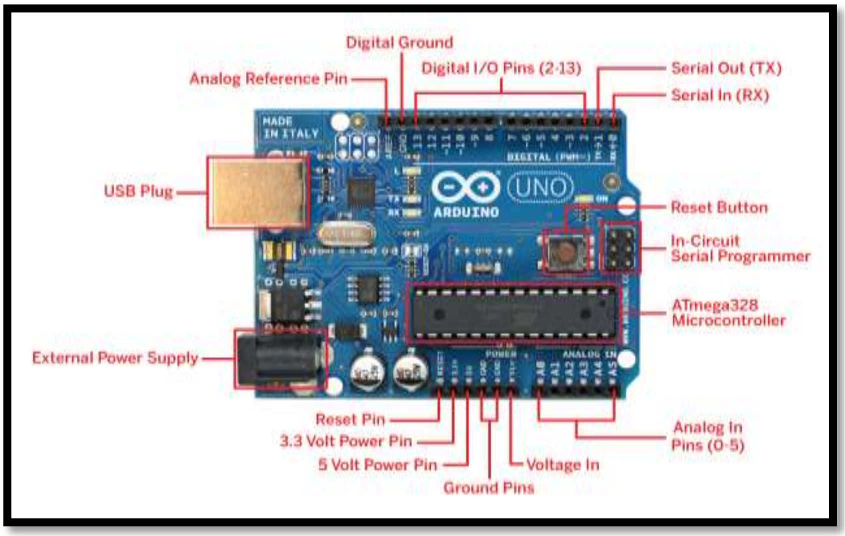

Figure 8.3 Arduino Uno

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Figure 8.4 Specifications of ATmega328P

2) ***DC Motor***

A DC motor is any of a class of electrical machines that converts direct current electrical power into mechanical power. Motor having sufficient torque to rotate a door is used. Now after developing required circuit we need to establish a serial communication with MATLAB.

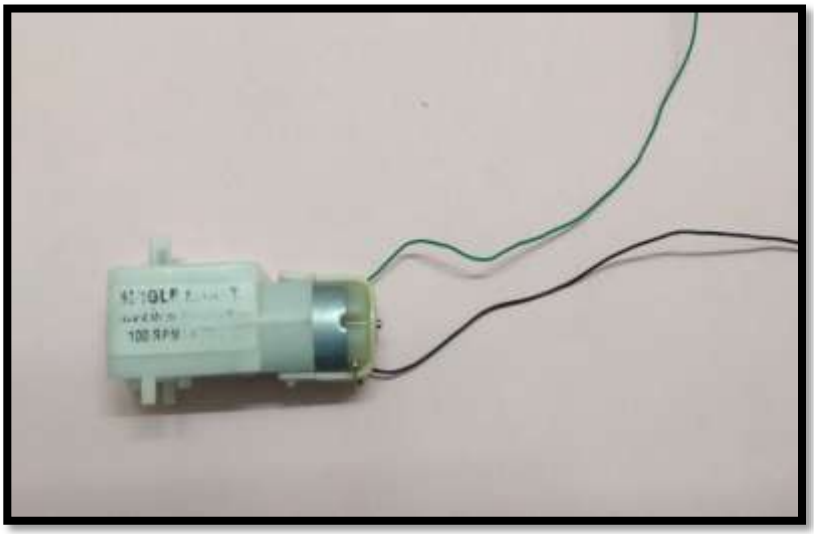

Figure 8.5 DC Motor

3) ***Motor Shield***

Design of a motor shield

- 2 connections for 5V 'hobby' servos connected to the Arduino's high-resolution dedicated timer
- Up to 4 bi-directional DC motors with individual 8-bit speed selection (so, about 0.5% resolution)
- Up to 2 stepper motors (unipolar or bipolar) with single coil, double coil, interleaved or micro-stepping
- 4 H-Bridges: L293D chipset provides 0.6A per bridge (1.2A peak) with thermal shutdown protection, 4.5V to 25V
- Pull down resistors keep motors disabled during power-up
- Big terminal block connectors to easily hook up wires (10-22AWG) and power
- Arduino reset button brought up top
- 2-pin terminal block to connect external power, for seperate logic/motor supplies

Figure 8.6 Motor Shield

3) ***Jumper Wires***

They are used to establish connections between various pins.

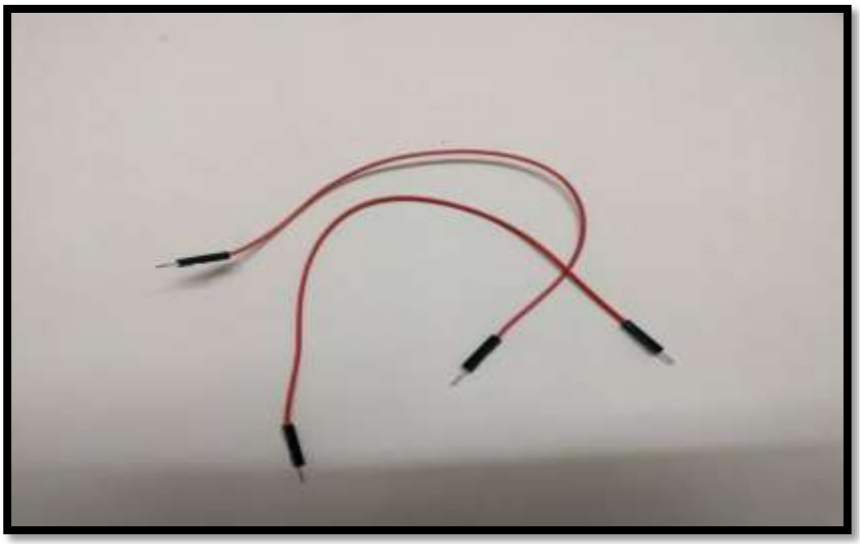

Figure 8.7 Jumper Wires

4) ***USB***

For transmitting data from MATLAB to Arduino board.

Figure 8.8 USB Cable

CHAPTER 9

IMPLEMENTATION

9.1. Establishing a serial communication between Arduino and MATLAB

MATLAB Support Package for Arduino:

MATLAB Support Package for Arduino (also referred to as "ArduinoIO Package") allows communicating with an Arduino Uno over a serial port. It consists of a MATLAB API on the host computer and a server program that runs on the Arduino. Together, they allow accessing Arduino analog I/O, digital I/O, operating servo motors, reading encoders, and even handling dc motors from the MATLAB command line.

***Steps to connect MATLAB –Arduino*:**

- For the Arduino:

1. Connect the circuit.
2. Connect the Arduino to PC.
3. From the ArduinoIO upload the adio.pde to arduino. Now arduino will be the server that will accept commands coming from MATLAB.

- For the MATLAB:

1. Run MATLAB as an administrator.
2. From the command window in MATLAB type "install_arduino" (making sure that the active directory is in ArduinoIO folder)
3. Create an M-File in MATLAB. Make sure to have the right COM port of Arduino.
4. Run the program.

9.2 Following the Detection/Recognition

If the face was recognized, the Arduino Board that was interfaced with MATLAB in Step 1, was used to run the motor and open the door *OR* If the person was not recognized, a buzzer was triggered.

9.3 Result

The person at the door is recognized and the name of the respected person appears on the command window and the door opens.

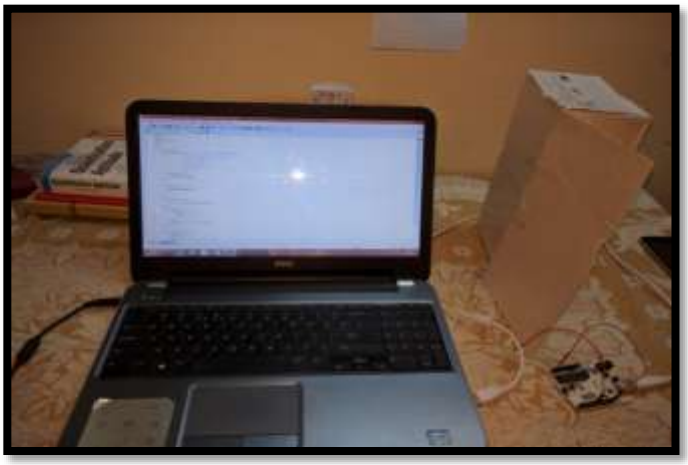

Figure 5.1 Hardware implementation of recognized face

2) The person on the door is not recognized, hence the door remains close and alarm rang.

Figure 9.2 Hardware implementation of unrecognized face

CHAPTER 10

CONCLUSION

10.1. Challenges

- The challenge faced during this project is that the recognition rate decreases for recognition under varying pose and illumination
- To cope with illumination distraction in practice, we discarded the first three eigen faces from the dataset. Since illumination is usually the cause behind the largest variations in face images, the first three eigen faces will mainly capture the information of 3-dimensional lighting changes, which has little contribution to face recognition. By discarding those three eigen faces, there will be a decent amount of boost in accuracy of face recognition.
- One of the limitations for Eigen face approach is in the treatment of face images with varied facial expressions and with glasses.

10.2. Applications

- To prevent robbery in highly secure areas with lesser power consumption and also help users for improvement of the door security of sensitive locations by using face detection and recognition.
- For helping people of old age who live alone at home for most part of the day since their children are working or not in town. This system helps them with their security without major movement required, on their part for it might be difficult for them. With the system already fed with data base of all the known people they would not be required to rush for the door bell every time it rang.
- The primary application of controlling the music arrangement of the room using facial expression recognition is to automatically give the desired setting to the room based on the expression recognized without any work on the person’s part.

10.3. Future Scope

Eigen face provides an easy and cheap way to realize face recognition:

- Its training process is completely automatic and easy to code.
- Recognition is simple and efficient compared to other matching approaches
- Eigen face adequately reduces statistical complexity in face image representation.
- No knowledge of geometry and reflectance of faces is required
- Once eigen faces of a database are calculated, face recognition can be achieved in real time.
- Eigen face can handle large databases.

REFERENCES

[1] Hiremath, Vinay, and Ashwini Mayakar. "Face recognition using Eigenface approach." *IDT workshop on interesting results in computer science and engineering, Sweden*. 2009.

[2] Delac, Kresimir, Mislav Grgic, and Sonja Grgic. "Independent comparative study of PCA, ICA, and LDA on the FERET data set." *International Journal of Imaging Systems and Technology* 15.5 (2005): 252-260.

[3] Lwin, Hteik Htar, Aung Soe Khaing, and Hla Myo Tun. "Automatic Door Access System Using Face Recognition."

[4] Bhardwaj, Ravikant, and Neeraj Gupta. "Face Recognition using Principal Component Analysis."

[5] Thuseethan, S., and S. Kuhanesan. "Eigenface Based Recognition of Emotion Variant Faces." *Available at SSRN 2752808* (2016).

[6] Sirovich, Lawrence, and Michael Kirby. "Low-dimensional procedure for the characterization of human faces." *Josa a* 4.3 (1987): 519-524.

[7] Kirby, Michael, and Lawrence Sirovich. "Application of the Karhunen-Loeve procedure for the characterization of human faces." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 12.1 (1990): 103-108.

[8] R. Cendrillon, B. Lovell (2000). "Real-time face recognition using eigenfaces" (PDF). *Visual Communications and Image Processing*. pp. 269–276. doi:10.1117/12.386642.

[9] Tiwari, Anamika. "Classification of Emotion from Facial Image Using Dimensionality Reduction Technique."

[10] Gosavi, Ajit P., and S. R. Khot. "Facial Expression Recognition Using Principal Component Analysis." International Journal of Soft Computing Engineering (IJSCE) ISSN (2013): 2231-2307.

[11] Slavkovic, Marijeta, and Dubravka Jevtic. "Face recognition using eigenface approach." Serbian Journal of electrical engineering 9.1 (2012): 121-130.