A Security System
based on
Face and Speech Authentication

A Report
Submitted in Partial Fulfillment of the Requirements
For the Degree of
Bachelor of Engineering
In
Electronics and Electrical Communications Engineering
By

Paramveer Singh Dhillon

PUNJAB ENGINEERING COLLEGE
DEEMED UNIVERSITY
CHANDIGARH, INDIA

Contents

List of Figures

Chapter 1

Introduction

Due to the rapid progress made in Artificial Intelligence and allied fields (e.g. Computer Vision, Machine Learning, Speech Processing, Natural Language Processing and Neural Networks) in the last three decades, it has been made possible to build autonomous intelligent systems.

One such motivation has been to build a security system based on face authentication. It is a very challenging task because the Face detection systems are very sensitive to:

- Pose Variation
- Illumination conditions
- Partial or full occlusions
- Image Orientation
- Facial Expressions

Next we consider the case of a speech authentication system. In this case problems faced are plenty and may outnumber those faced in Face Recognition. The main problems faced are due to omnipresent noise which may interfere with the sound signal, the pitch of different humans also vary greatly lastly speech processing systems are very sensitive to different accents and inflections used by people belonging to different demographic locations.

As a part of our senior year major project we have endeavored to build a security system based on Face and Speech Authentication. Security is embedded in two stages in the system. The face authentication system constitutes the first stage and speech authentication constitutes the second stage.

In the first stage the system takes the image of the face of the user, if user is one of the persons who is allowed to enter the system then the face authentication system gives the "green signal" (turns on green LED) and the control of the system passes to the second stage of the system i.e. the speech authentication system.

In the second stage the user who has passed the first level of security i.e. the face authentication system, is required to speak a password **(any word from the List 1 given on Page 20)** within a certain time frame, if the password is correct then the user has passed the second authentication also and a second "green signal" is given to imply that the user is one of the "authorized" persons to enter the system.

In all other cases i.e. the cases in which the user

- Fails at first level of authentication
- Passes at First Level and Fails at Second Level

a “red signal” (turns on red LED) is given implying that the user has failed the authentication system.

The literature survey shows that the security systems implemented till date have either used face or speech authentication and not both.

The reported efficiency of the system is ~85%, with the efficiency of face authentication system being ~90% and efficiency of speech authentication system being ~95%.

The theoretical and implementation details of each authentication system are given in succeeding sections separately.

Chapter 2

Face and Speech Authentication Software Modules

As mentioned earlier, the security system consists of authentication at two stages, i.e. Face Authentication and Speech Authentication, with Face Authentication preceding Speech Authentication. It is only after clearing the first level of authentication that the user proceeds to the second level. Failing the first level the authentication fails and the user is identified as an 'unauthorized' person. The details of the two subsystems are given in the following sections.

2.1 Face Authentication Module

This module is the first and the most important subsystem in the entire Security System and the performance of the entire system is dependent on the performance of this section. In this module we detect the face of the user and match it with a database of faces using certain transformations and if the face matches with a 'training sample' i.e. it matches with a face in the database upon using certain transformations then the authentication succeeds and the user proceeds to the speech authentication part otherwise the user is not authorized to enter the system.

For face detection we have used "**EigenFaces**" approach as postulated by *Turk and Pentland.* The "**EigenFaces**" approach is explained in detail in succeeding sections. In short, this approach is based on using PCA (Principal Component Analysis) for dimensionality reduction of the feature space consisting of faces of various persons, in this case, the faces of various persons who are to be allowed access. After applying PCA to the data we do some post processing by using various smoothing operators like Laplacian operator etc. and then we find an estimated 'mean image' from the given data.

Now, when we have the image of user, i.e. the query image, we subtract it from the mean image and find a match in the database. If the input image matches the reconstructed image then a match is reported and the authentication succeeds.

In our case we have constructed a database in form of a .MAT file (**MATLAB™)** by taking about 200 normalized RGB images of size (320*240). The results of the Face

Authentication system alone are in the range of ~85-90 % and they vary greatly with illumination changes and facial expression changes which is in fact the drawback of any Face Recognition system.

In the next section we discuss the State of the Art in Face Detection, problems faced in Face Detection systems and lastly we explore the theoretical base of **EigenFaces** approach that we have used in our recognition system.

2.1.1 State of the Art in Face Recognition

As mentioned earlier Face recognition is a problem which is one of the most researched problems in Computer Vision and Artificial Intelligence.

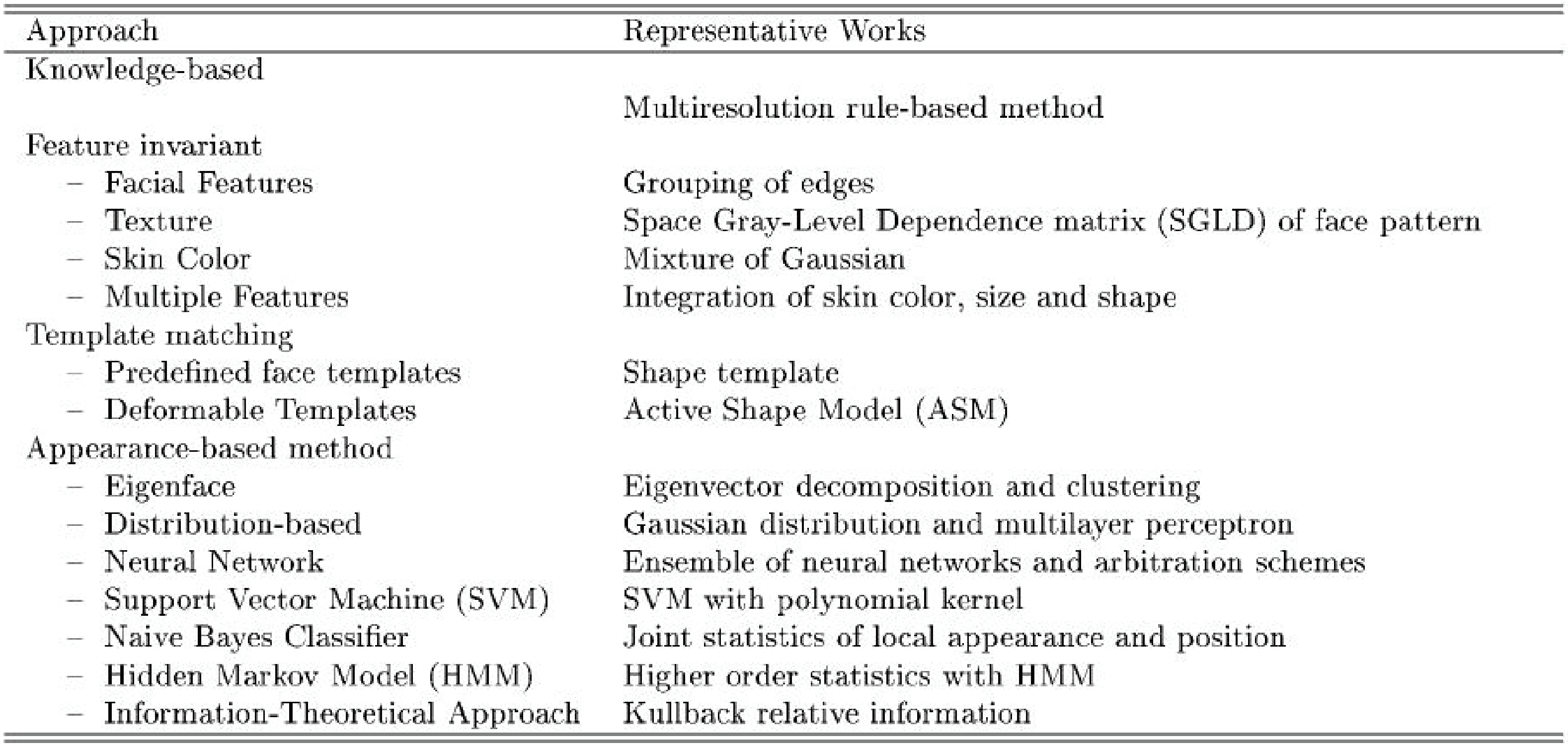
In this section, we review existing techniques to detect faces from a single intensity or color image. Single image detection methods can be classified into four categories

1. **Knowledge-based methods:** These rule-based methods encode human knowledge of what constitutes a typical face. Usually, the rules capture the relationships between facial features. These methods are designed mainly for face localization.

2. **Feature invariant approaches:** These algorithms aim to find structural features that exist even when the pose, viewpoint, or lighting conditions vary, and then use these to locate faces. These methods are designed mainly for face localization.

3. **Template matching methods:** Several standard patterns of a face are stored to describe the face as a whole or the facial features separately. The correlations between an input image and the

TABLE 1
Categorization of Methods for Face Detection in a Single Image



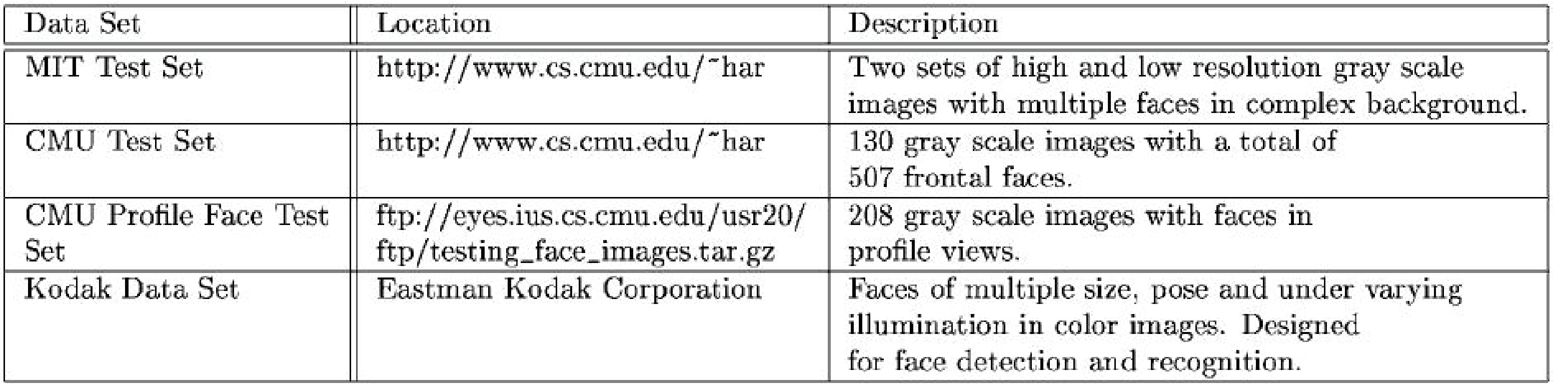
Approach	Representative Works
Knowledge-based	
	Multiresolution rule-based method
Feature invariant	
– Facial Features	Grouping of edges
– Texture	Space Gray-Level Dependence matrix (SGLD) of face pattern
– Skin Color	Mixture of Gaussian
– Multiple Features	Integration of skin color, size and shape
Template matching	
– Predefined face templates	Shape template
– Deformable Templates	Active Shape Model (ASM)
Appearance-based method	
– Eigenface	Eigenvector decomposition and clustering
– Distribution-based	Gaussian distribution and multilayer perceptron
– Neural Network	Ensemble of neural networks and arbitration schemes
– Support Vector Machine (SVM)	SVM with polynomial kernel
– Naive Bayes Classifier	Joint statistics of local appearance and position
– Hidden Markov Model (HMM)	Higher order statistics with HMM
– Information-Theoretical Approach	Kullback relative information

stored patterns are computed for detection. These methods have been used for both face localization and detection.

4. **Appearance-based methods:** In contrast to template matching, the models (or templates) are learned from a set of training images which should capture the representative variability of facial appearance. These learned models are then used for detection. These methods are designed mainly for face detection.

Next we show experimental results for face detection and recognition of various face recognition algorithms that are widely used. The results were conducted on various standard face test sets as shown below:

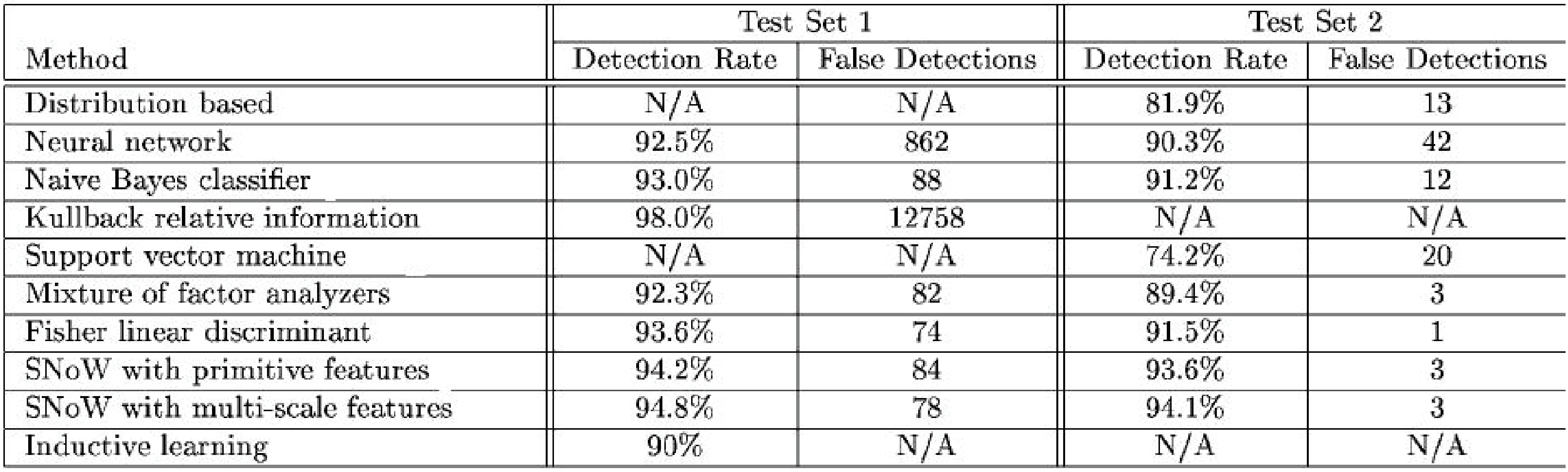
Test Sets for Face Detection



Data Set	Location	Description
MIT Test Set	http://www.cs.cmu.edu/˜har	Two sets of high and low resolution gray scale images with multiple faces in complex background.
CMU Test Set	http://www.cs.cmu.edu/˜har	130 gray scale images with a total of 507 frontal faces.
CMU Profile Face Test Set	ftp://eyes.ius.cs.cmu.edu/usr20/ftp/testing_face_images.tar.gz	208 gray scale images with faces in profile views.
Kodak Data Set	Eastman Kodak Corporation	Faces of multiple size, pose and under varying illumination in color images. Designed for face detection and recognition.

The corresponding results are shown below:

Experimental Results on Images from Test Set 1 (125 Images with 483 Faces) and Test Set 2 (23 Images with 136 Faces)



	Test Set 1		Test Set 2	
Method	Detection Rate	False Detections	Detection Rate	False Detections
Distribution based	N/A	N/A	81.9%	13
Neural network	92.5%	862	90.3%	42
Naive Bayes classifier	93.0%	88	91.2%	12
Kullback relative information	98.0%	12758	N/A	N/A
Support vector machine	N/A	N/A	74.2%	20
Mixture of factor analyzers	92.3%	82	89.4%	3
Fisher linear discriminant	93.6%	74	91.5%	1
SNoW with primitive features	94.2%	84	93.6%	3
SNoW with multi-scale features	94.8%	78	94.1%	3
Inductive learning	90%	N/A	N/A	N/A

So, it can be concluded that face recognition is not a solved problem as none of the algorithms is totally perfect i.e. Hit-Ratio is not 100% for any of the algorithms, as can be seen above. Some algorithms perform better in certain conditions and the other performs well in different conditions. So, while choosing a face recognition algorithm we have to take into consideration various external factors like illumination conditions, background clutter etc.

2.1.2 Problems Encountered in Face Recognition

Face Detection is one of the most exciting problems encountered in Computer Vision. Furthermore it is one of the most complex and intriguing problems to handle. Face detection systems form the heart of many intelligent vision-based human computer interaction systems, state of the art video surveillance systems implemented on various airports throughout the world and various security systems based on face authentication alone

The problem of face recognition is very challenging because faces are non-rigid and have a high degree of variability in size, shape, color and texture. More precisely the challenges associated with face detection are due to

- **Pose:** The images of a face vary due to the relative camera-face pose (frontal, 45 degree, profile, upside down), and some facial features such as an eye or the nose may become partially or wholly occluded.

- **Presence or absence of structural components**: Facial features such as beards, mustaches, and glasses may or may not be present and there is a great deal of variability among these components including shape, color, and size.

- **Facial expression**: The appearances of faces are directly affected by a person's facial expression.

- **Occlusion**: Faces may be partially occluded by other objects. In an image with a group of people, some faces may partially occlude other faces.

- **Image orientation:** Face images directly vary for different rotations about the camera's optical axis.

- **Imaging conditions:** When the image is formed, factors such as lighting (spectra, source distribution and intensity) and camera characteristics (sensor response, lenses) affect the appearance of a face.

The above effects can be illustrated in a better way by considering the following images, it shows how varying illumination conditions and varying Facial Expressions, drastically change the actual image and its content i.e. face in this case

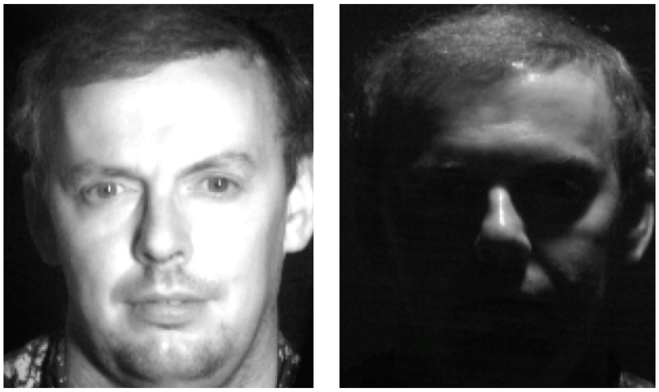


Fig. 1: Same image in two different illumination conditions (drastic change)

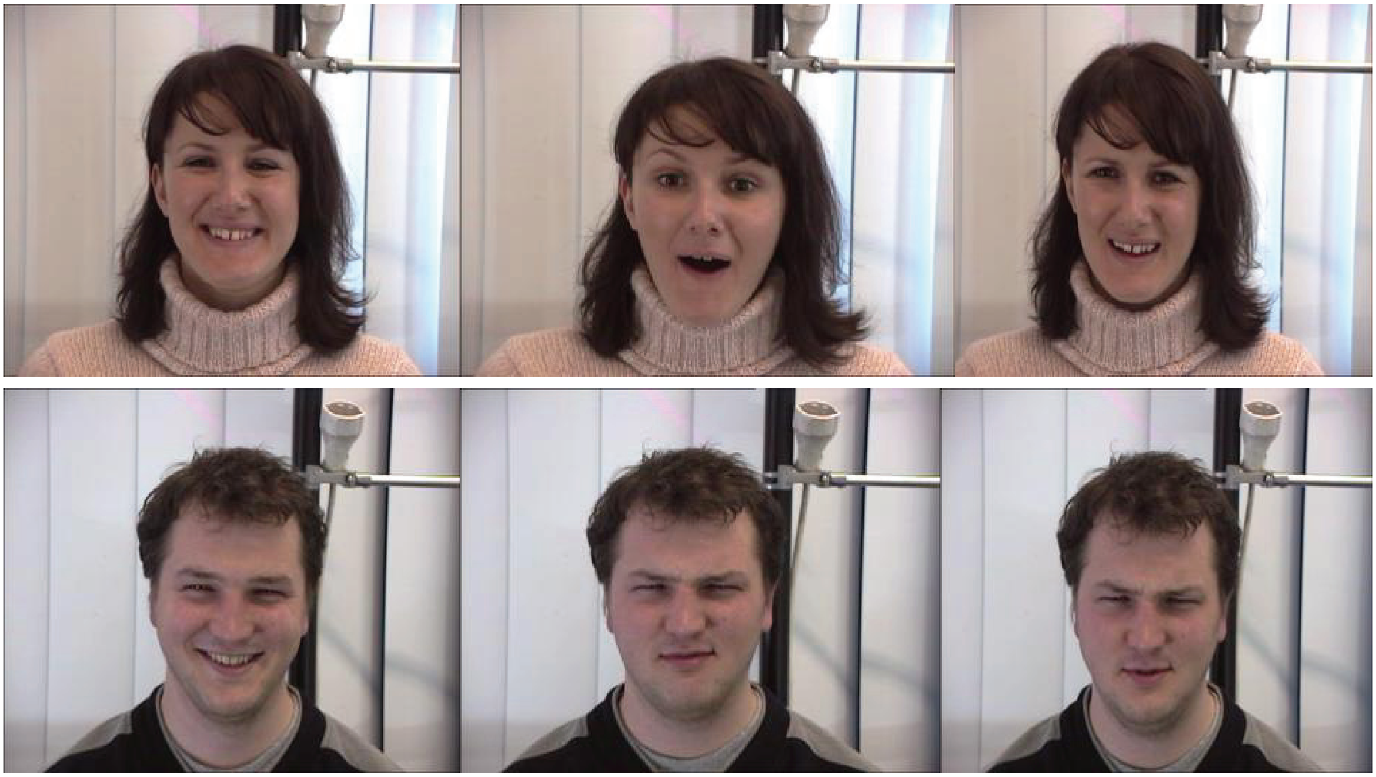


Fig. 2: Images of two persons with varying facial expressions

2.1.3 **EigenFace** approach to Face Recognition

We have used EigenFace approach in our system to detect and authenticate the faces. This approach was developed by Matthew Turk as part of his doctoral thesis at Massachusetts Institute of Technology. It was one of the first successful approaches for face recognition and it showed good results. Significant improvements have been made in this approach over the years.

This approach treats the face recognition problem as intrinsically two dimensional (2-D) problem rather than requiring recovery of 3-D geometry. This approach takes advantage of the fact that faces are normally upright and thus may be described by a small set of 2-D characteristic views. The algorithm functions by projecting face images onto a feature space that spans the significant variations among known face images. The significant features are known as "EigenFaces", because they are the eigenvectors (principal components) of the set of faces and normally do not necessarily correspond to features like eyes, ears, nose. The projection operation characterizes an individual face by a weighted sum of eigenface features, so to recognize a particular face it is necessary only to compare these weights to those of known individuals. Some particular advantages of this approach are that it provides the ability to learn and later recognize new faces in an unsupervised manner, and that it is easy to implement using neural network architecture.

Our system for face recognition based on **EigenFaces** uses the following four sets of images to create the eigen space for implementation of the EigenFace Algorithm

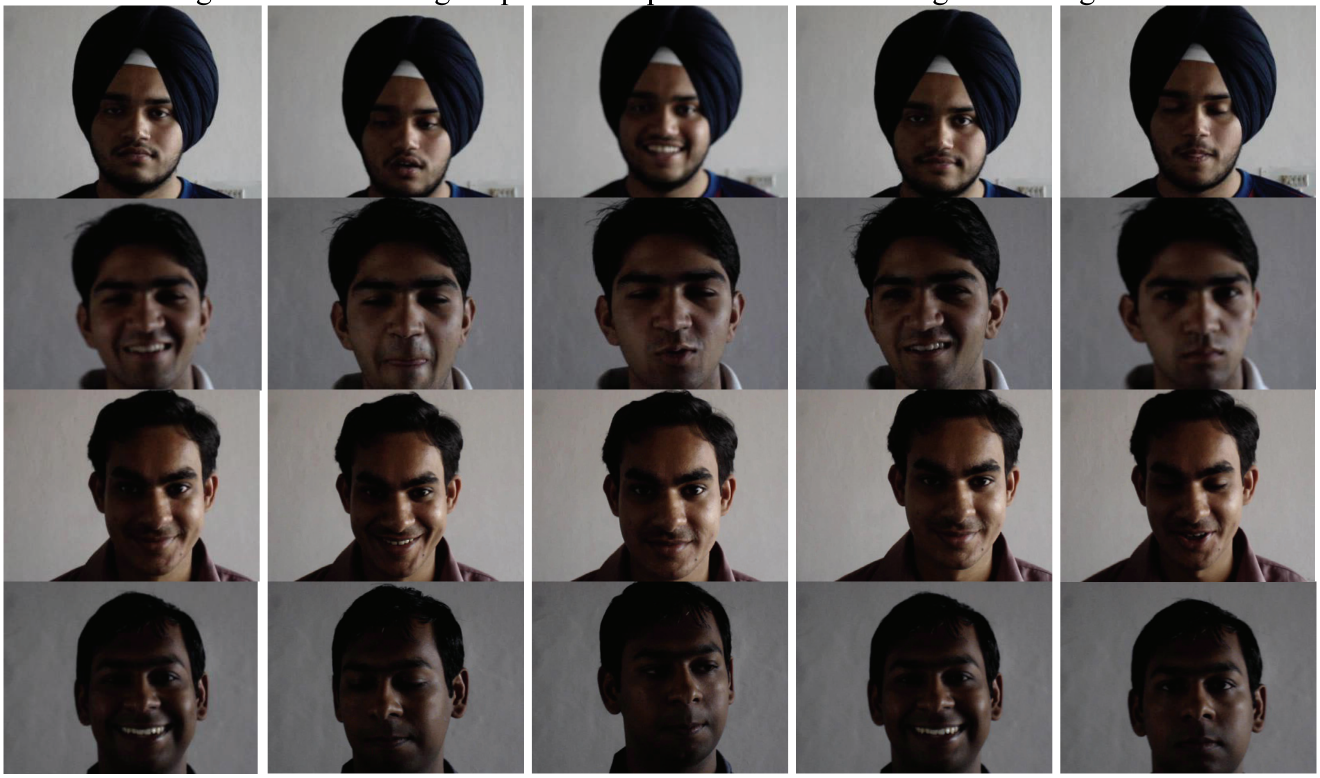


Fig. 3: A subset of the training images used to create Eigen Space for Face Recognition

Flowchart for Building Eigen-Space and later using it for Face-Authentication is as below:

1. The first step is to obtain a set S with M face images. In our example M = 190 as shown at the beginning of the tutorial. Each image is transformed into a vector of size N and placed into the set.

$$S = \{ \Gamma_1, \Gamma_2, \Gamma_3, \ldots\ldots\ldots\ , \Gamma_M \}$$

2. After you have obtained your set, you will obtain the mean image **Ψ**

$$\Psi = \frac{1}{M} \sum_{n=1}^{M} \Gamma_n$$

3. Then you will find the difference **Φ** between the input image and the mean image

$$\Phi_i = \Gamma_i - \Psi$$

4. Next we seek a set of M orthonormal vectors, $\mathbf{u}_n$**,** which best describes the distribution of the data. The k^th vector, $\mathbf{u}_k$**,** is chosen such that

$$\lambda_k = \frac{1}{M} \sum_{n=1}^{M} \left(u_k^T \Phi_n \right)^2$$

is a maximum, subject to

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1 & \text{if } l = k \\ 0 & \text{otherwise} \end{cases}$$

Note: $\mathbf{u}_k$ and $\boldsymbol{\lambda}_k$ are the eigenvectors and eigenvalues of the covariance matrix **C**

5. We obtain the covariance matrix **C** in the following manner

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T \qquad\qquad A = \{ \Phi_1, \Phi_2, \Phi_3, \ldots\ldots\ , \Phi_n \}$$

$$= AA^T$$

6. A^T

$$L_{mn} = \Phi_m^T \Phi_n$$

7. Once we have found the eigenvectors, **v$_l$**, **u$_l$**

$$u_l = \sum_{k=1}^{M} v_{lk}\Phi_k \qquad l = 1, \ldots\ldots, M$$

Face Recognition Procedure

1. A new face is transformed into its eigenface components. First we compare our input image with our mean image and multiply their difference with each eigenvector of the L matrix. Each value would represent a weight and would be saved on a vector **Ω**.

$$\omega_k = u_k^T(\Gamma - \Psi) \qquad \Omega^T = [\omega_1, \omega_2, \ldots\ldots, \omega_M]$$

2. We now determine which face class provides the best description for the input image. This is done by minimizing the Euclidean distance

$$\varepsilon_k = \|\Omega - \Omega_k\|^2$$

3. The input face is considered to belong to a class if ε_k is below an established threshold θ_ε. Then the face image is considered to be a known face. If the difference is above the given threshold, but below a second threshold, the image can be determined as a unknown face. If the input image is above these two thresholds, the image is determined NOT to be a face.

4. If the image is found to be an unknown face, you could decide whether or not you want to add the image to your training set for future recognitions. You would have to repeat steps 1 trough 7 to incorporate this new face image.

2.2 Speech Authentication Module

Voice Recognition is a fascinating field spanning several areas of computer science and mathematics. Reliable speech recognition is a hard problem, requiring a combination of many techniques; however modern methods have been able to achieve an impressive degree of accuracy. This module attempts to examine those techniques, and to apply them to build a simple voice recognition system. It is designed to be able to recognize a vocabulary of 20 words, spoken individually. The method used is a simple one, involving a simple count of the frequency of zero crossings, but it is quite applicable to the voice recognition problem in general.

The major challenge in voice/speech recognition is due to interference of omnipresent noise which distorts the actual sound waveform and prevents the recognition of the actual word. Also the effects of demographically varying accents and inflections preclude the efficiency of the system in a major way.

2.2.1 General Approach used for Speech Recognition

The process of voice recognition is typically divided into several well defined steps. Different systems vary on the nature of theses steps, as well as how each step is implemented, but the most successful systems follow a similar methodology.

1. Divide the sound wave into evenly spaced blocks
2. Process each block for important characteristics, such as strength across various frequency ranges, number of zero crossings, and total energy.
3. Using this characteristic vector, attempt to associate each block with a phone, this is the most basic unit of speech, producing a string of phones.
4. Find the word whose model is the most likely match to the string of phones which was produced.

Step 2 typically involves performing a spectrum analysis of the block. This can be done with a Fast Fourier Transform (FFT), or with a bank of frequency filters, but the most successful technique to date has been that of Linear Predictive Coding. Additional important features include analyzing the total energy, the change in the features over time, and the number of zero crossings. Step 3 is often done via a decision tree. Each phone often has very prominent characteristics which narrow the field of consideration. Additional characteristics then separate similar sounding phones. The final decisions are often mistaken, and these mistakes must be accounted for later. Step 4 has been implemented with a high degree of success using Hidden Markov Models (HMM's). A HMM is constructed for each word in the vocabulary, and then the string of phones is compared against each HMM, to determine which model is the most likely match

The various steps mentioned above are described below in context to our speech authentication system.

Speech Detection

The first task is to identify the presence of a speech signal. This task is easy if the signal is clear, however frequently the signal contains background noise, resulting from a noisy microphone, a fan running in the room, etc. The signals obtained were in fact found to contain some noise. We have used two criteria to identify the presence of a spoken word. First, the total energy is measured, and second the number of zero crossings are counted. Both of these were found to be necessary, as voiced sounds tend to have a high volume (and thus a high total energy), but a low overall frequency (and thus a low number of zero crossings), while unvoiced sounds were found to have a high frequency, but a low volume. Only background noise was found to have both low energy and low frequency. The method was found to successfully detect the beginning and end of the several words tested. Note that this is not sufficient for the general case, as fluent speech tends to have pauses, even in the middle of words (such as in the word 'acquire', between the 'c' and 'q'). In fact reliable speech detection is a difficult problem, and is important part of speech recognition; however the method we described is sufficient for this project.

Blocking

The second task is blocking. Older speech recognition systems first attempted to detect where the phones would start and finish, and then block the signal by placing one phone in each block. However, phones can blend together in many circumstances, and this method generally could not reliably detect the correct boundaries. Most modern systems simply separate the signal into blocks of a fixed length. These blocks tend to overlap, so that phones which cross block boundaries will not be missed. This project uses blocks which are 30 msec in length (containing

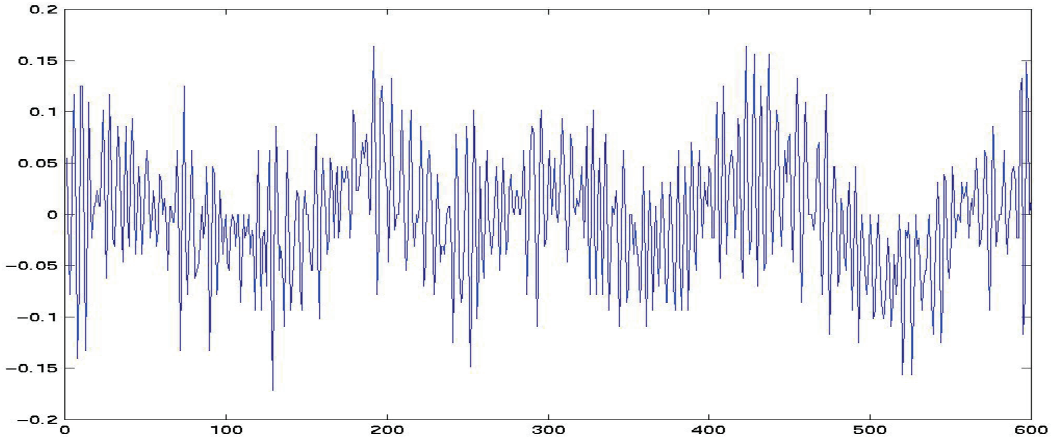


Fig. 4 : A typical block for **'S'** phone

600 samples), and which shift by 10 msec increments.

Obtaining a frequency spectrum

The next important step in the processing of the signal is to obtain a frequency spectrum of each block. The information in the frequency spectrum is often enough to identify the phone. The purpose of the frequency spectrum is to identify the *formants*, which are the peaks in the frequency spectrum. Vowels are often uniquely identified by their first two formants. This experiment has shown that the identification of formants is not a trivial task. One method to obtain a frequency spectrum is to apply an FFT to each block. The resulting information can be examined manually to find the peaks, but it is quite noisy, which makes the take difficult for a computer to identify the peaks. Very useful data can still be obtained. This is often done by measuring the strength across various frequency ranges. The following image show the data returned from an FFT:

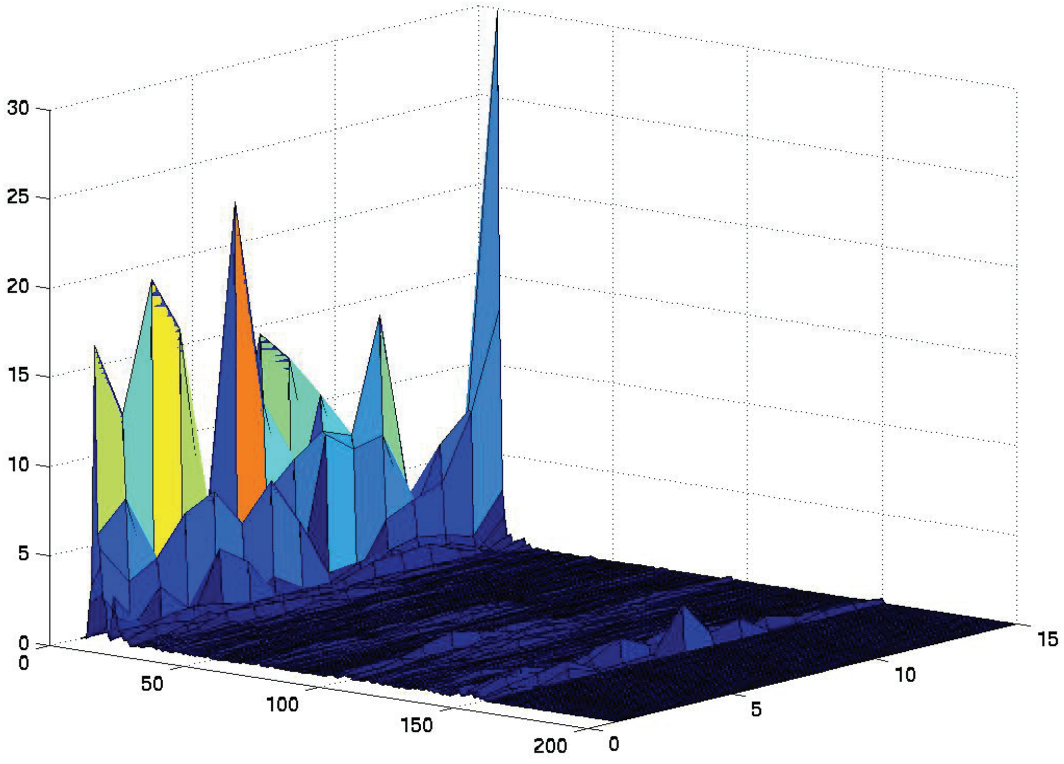


Fig. 5: FFT Plot of sound signal

2.2.2 Vocabulary of our system and implementation details

In our system we have implemented Speech Recognition by differentiating unvoiced fricatives (Unvoiced fricatives are 'f' 's' 'sh' and 'th' (as in think)) from voiced phones. A high number of zero crossings is indicative of the presence of an unvoiced fricative (150-300 zero crossings per block are typical), whereas voiced phones tend to have much fewer zero crossings (around 10-50 per block).

1. First the sound wave is divided into overlapping blocks. Each block contains 600 samples which is about 30 msec long.

2. Next the blocks are examined to see if they contain any significant data. The total energy and the number of zero crossings are both measured to determine this.

3. Finally, if there is a substantial signal, the zero crossings are again counted in each block.

If we know the word spoken is a word from the list on **Page 20(below)**, we can simply look for unvoiced fricatives, by looking for blocks with sufficient numbers of zero crossings. The presence of an unvoiced fricative would indicate that the word being spoken is from List 1, while the absence of such would indicate that the word spoken is from List 2

Vocabulary of our System

List1 (Suggested Passwords for the System)	List 2
recherché	cool
think	umbro
church	drool
shut	number
guess	hole
fat	zebra
yes	no
Champs Elysees	zoo
Abhishek	Paramveer
Anshul	Rajan

Chapter 3

Hardware Module and PC Parallel Port Interface

The Hardware module consists of the Circuitry and PC parallel port interface. The Circuitry was used to decode the authentication information sent through the parallel port and accordingly actuate a DC motor if “Access Granted” and glow a “red” LED if “Access Denied”

3.1 Hardware Module

3.1.1 Circuit Diagram

The circuit diagram schematic is as shown on the figure on next page.

3.1.2 Specifications of Components used

The data sheets of various components are attached at the end of the report.

3.2 PC Parallel Port Interface

In our security system we have used a parallel port to interface the hardware with the software. There are many reasons for choosing Parallel Port over other communication ports.

Firstly, parallel port is a reliable mode as there are less chances of data loss, secondly it allows us to send various bits of data simultaneously. Thirdly there is built in support for Parallel Port in ***MATLAB***™. Fourthly it is easier to program and less complex decoding circuitry at receiver end.

3.2.1 Specifications of PC Parallel Port

Parallel port is a simple and inexpensive tool for building computer controlled devices and projects. The simplicity and ease of programming makes parallel port popular. The parallel port is often used in Computer controlled robots, Atmel/PIC programmers, home automation etc.

Parallel port modes

The IEEE 1284 Standard which has been published in 1994 defines five modes of data transfer for parallel port. They are,

1) Compatibility Mode
2) Nibble Mode
3) Byte Mode
4) EPP
5) ECP

Hardware

The pin outs of DB25 connector is shown in the figure below

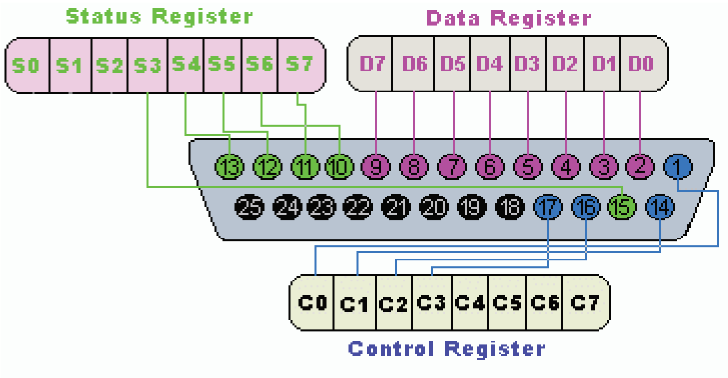


Fig. 6: Parallel Port DB 25 connector and its layout

The lines in DB25 connector are divided in to three groups, they are

1) Data lines (data bus)
2) Control lines
3) Status lines

As the name refers , data is transferred over data lines , Control lines are used to control the peripheral and the peripheral returns status signals back computer through status lines. These lines are connected to Data, Control and Status registers internally. The details of parallel port signal lines are given below

Pin No (DB25)	**Signal name**	**Direction**	**Register - bit**	**Inverted**
1	nStrobe	Out	Control-0	Yes
2	Data0	In/Out	Data-0	No
3	Data1	In/Out	Data-1	No
4	Data2	In/Out	Data-2	No
5	Data3	In/Out	Data-3	No
6	Data4	In/Out	Data-4	No
7	Data5	In/Out	Data-5	No
8	Data6	In/Out	Data-6	No
9	Data7	In/Out	Data-7	No
10	nAck	In	Status-6	No
11	Busy	In	Status-7	Yes
12	Paper-Out	In	Status-5	No
13	Select	In	Status-4	No
14	Linefeed	Out	Control-1	Yes
15	nError	In	Status-3	No
16	nInitialize	Out	Control-2	No
17	nSelect-Printer	Out	Control-3	Yes
18-25	Ground	-	-	-

Fig. 7 : Details of parallel port signal lines

3.2.2 Parallel Port support in ***MATLAB***™

MATLAB has built in support for parallel port and we only need to create an object by instantiating the *digitalio()* function of MATLAB Data Acquisition Toolbox. For eg:

parport = digitalio('parallel','LPT1');

Now we can use the '*parport*' object anywhere in a program to control the parallel port or send and receive data.
We can add lines to the parallel port object as follows

hwlines = addline(dio,0:7,'out');

The above command configures the eight data-pins (D0-D7) as output pins i.e. the parallel port is configured as an output port.
Similarly we can configure the port as input port or we may configure some pins as input pins and others as output pins.

The parallel port follows TTL Logic to define voltage standards i.e. a voltage in the range of (2.7V-5V) is a logical 1 and (0V-2.7V) is a logical 0.

In our system we have used data pins D0 and D1 for Face and Speech Authentication Data and we have grounded Pin-18 of the parallel port.

Chapter 4

Results and Conclusions

The System was run on a computer with 1.7 GHz processor clock speed, 256 MB RAM and 1 GB Virtual/Paging Memory. Operating system used was Microsoft© Windows XP™. Image acquisition device used was a simple Webcam with 320X240, 30 fps and 640X480, 15fps specifications. We used the 320X240 resolution.

Results shown by the system were quite encouraging, we ran 35 trials of the system as whole and also 35 trials of Face and Speech Authentication Modules Separately. The results have been shown graphically as on next page. As can be seen the performance of the entire system is around ~80% mark. The performance of the Speech Recognition System is very good ~90%.

The trials were run in varying environments like different illumination conditions, presence of noise, and different facial expressions.

The results can be improved by using a better resolution camera which can offset the effects of illumination conditions to some extent and also the results can be improved if we use a dedicated system to perform the authentication i.e. a System on Chip as it can process real-time information faster than MATLAB on a personal computer.

We can conclude form the graphs on the next page that speech authentication system is quite sensitive to noise as can be seen by the drop in hit-ratio at 20-25 trials. During these trials we introduced some noise to see its effect on the speech module.

The face recognition module also shows lesser efficiency at initial stages compared to later stages. This can be justified by the fact that the webcam used for image acquisition needs some time to settle down before it starts to give desirable results.

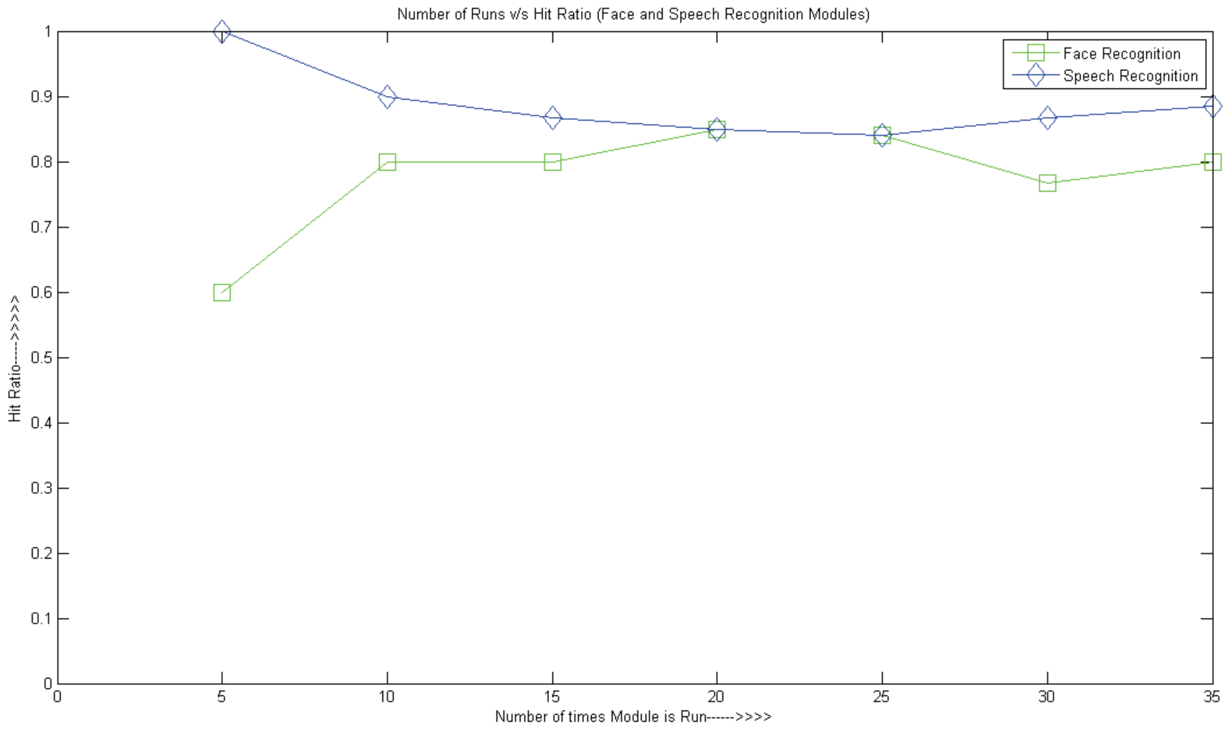


Fig. 8: Graph showing the performance of Face and Speech Recognition Modules

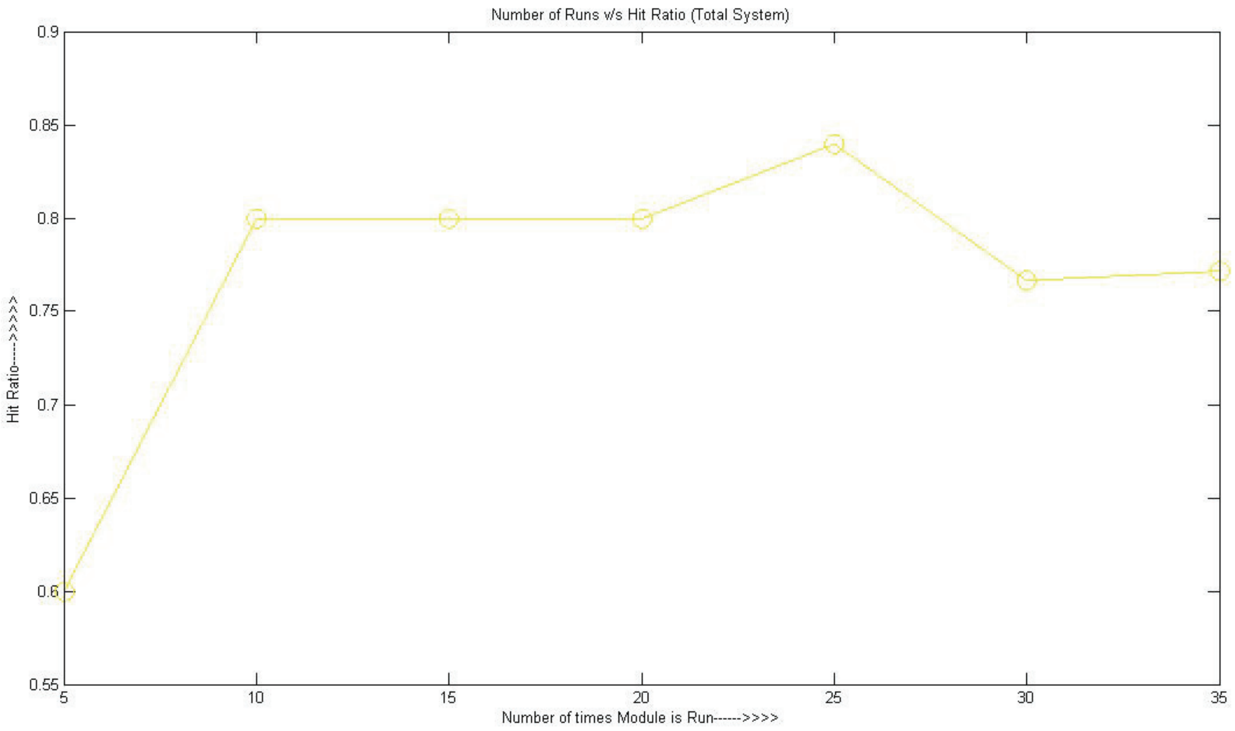


Fig. 9: Graph showing the performance of the complete System

Appendix

A. Source Codes (8051µC/MATLAB)

8051µC source code

The assembly language source code for 8051µC is as given below. This code is the main subroutine for processing the data coming from parallel port. It contains the logic for glowing ‘red’ LED when either Face or Speech Authentication fails and ‘green’ LED when they succeed.

```
ORG 00h
START : MOV P3,#0FFH
             MOV P1,#00H
             MOV P2,#00H
             NOP
             MOV P2,#50H
HERE :   JB P3.0,HERE
LOOP :   JNB P3.0,LOOP
      MOV P2,#90H
              MOV A,#09H
DISPLAY : MOV P1,A
          LCALL DELAY
          LCALL DELAY
          LCALL DELAY
HERE1:   JB P3.1,HERE1
         JB P3.1,MOTOR
BACK :    DEC A
          JNZ DISPLAY
          LJMP START
MOTOR : MOV P2,#0A1H
                LCALL DELAY
                LCALL DELAY
                MOV P2,#0A2H
                LCALL DELAY
                LCALL DELAY
                LJMP START
  ORG 100H
DELAY : MOV R1,#03H
DELAY1: MOV R2,#0FFH
DELAY2: MOV R3,#0FFH
HERE1 : DJNZ R3,HERE1
```

```
          DJNZ R2,DELAY2
          DJNZ R1,DELAY1
           RET
        END
```

MATLAB Source Code

The code below initializes the parallel port and webcam. Then it captures the user's image and processes it to find if it is a known face. If it is a known face, in that case, the speech authentication module is called and second level of authentication begins, else authentication fails.

```
clear all
close all
clc
dio = digitalio('parallel','LPT1');
hwlines = addline(dio,0:7,'out');
putvalue(dio,0);
disp('Port Initialized');
load faces;
vidobj = videoinput('winvideo', 1);
preview(vidobj);
pause;
delete(vidobj);
vidobj = videoinput('winvideo', 1);
snapshot = getsnapshot(vidobj);
snapshot=rgb2gray(snapshot);
imwrite(snapshot,'1.jpg');
disp('Thanks!!')
InputImage = imread(strcat('C:\Project\','1.jpg'));
InImage=reshape(double(InputImage)',irow*icol,1);
temp=InImage;
me=mean(temp);
st=std(temp);
temp=(temp-me)*ustd/st+um;
NormImage = temp;
Difference = temp-m;
InImWeight = [];
for i=1:size(u,2)
t = u(:,i)';
WeightOfInputImage = dot(t,Difference');
InImWeight = [InImWeight; WeightOfInputImage];
end
ll = 1:M;
% Find Euclidean distance
e=[];
for i=1:size(omega,2)
q = omega(:,i);
DiffWeight = InImWeight-q;
mag = norm(DiffWeight);
```

```
e = [e mag];
end
kk = 1:size(e,2);
MaximumValue=max(e); % maximum euclidian distance
Avg=mean(e);
MinimumValue=min(e);    % minimum euclidian distance
if(Avg >22900)
      display('Authentication Failed!!');
      display('Access Denied');
else
      putvalue(dio,1);
      display('Speak your password');
      yesno;
end
```

This second module of code deals with speech authentication and grants access if the authentication succeeds

```
% Block Size
N = 600;

% Block Increment
M = 200;

% Threshold to discard
sumthresh = 0.035;
zerocrossthresh = 0.060;

ai = analoginput('winsound');
addchannel(ai,1);
Fs = 8000;
duration = 5;
set (ai, 'SampleRate', Fs);
set (ai, 'SamplesPerTrigger', duration*Fs);

%% Start the acquisition and retrieve the data
display('Start');
start(ai);
rawdata = getdata(ai);



blockeddata = block(rawdata,N,M);
strippeddata = strip(blockeddata,sumthresh,zerocrossthresh);
zerocrossdata = zerocrossmap(strippeddata);

plot(zerocrossdata);
```

```
maxzerocross = max(zerocrossdata);

if (maxzerocross < 5)
      f='INSUFFICIENT DATA';
      elseif (maxzerocross > 120)
      display('Aceess Granted!!');
      putvalue(dio,2);
else
      display('Authentication Failed!!');
      display('Access Denied');
end
```

B. Cost Analysis

Component Used	Price (in Rs.)
89S51 (Microcontroller)	40
ULN 2003 (Driver for Motor)	12
74LS244 (Octal Buffer)	20
D.C Motor	40
4 General Purpose PCBs	45
LEDs	5
Resistances (470Ω ± 5%)	5
Webcam	980
Microphone	120
Connecting Wires (Ribbon Wire)	15
Power Supply (Total Cost)	100

Total Cost: 1382 Rs.

References

[1]. Ming Hsuan Yang, David Kriegman, Narendra Ahuja, "*Detecting Faces in Images: A Survey*", IEEE Transactions on PAMI, Vol. 24, No.1, January 2002.

[2]. W. Zhao, Rama Chellappa, P.J. Phillips, A. Rosenfeld, *"Face Recognition: A Literature Survey",* ACM Computing Surveys, Vol.35, No.4, December 2003.

[3]. http://www.face-rec.org

[4]. Matthew Turk and Alex Pentland, *"Face recognition using eigenfaces",* Proc. IEEE Conference on Computer Vision and Pattern Recognition (1991): 586–591.

[5]. Rafael C.Gonzalez, Richard E. Woods, Steven L. Eddins, "*Digital Image Processing using MATLAB™",* Pearson Education, 2004

[6].Julius Open Source Speech Recognition Engine,
http://julius.sourceforge.jp/en_index.php?q=en/index.html

[7]. http://www.alldatasheet.com/